fare elettronica

CULTURA ELETTRONICA APPLICATA

SETTEMBRE
n° 243 • Anno 21

Economica CNC a 3 assi

COME NASCE UN CIRCUITO INTEGRATO

MIKROBASIC
PWM con il PIC

GENERATORE EOLICO
Completamento e messa in opera

ALIMENTATORI SWITCHING
Switching flyback e l'isolamento ingresso-uscita

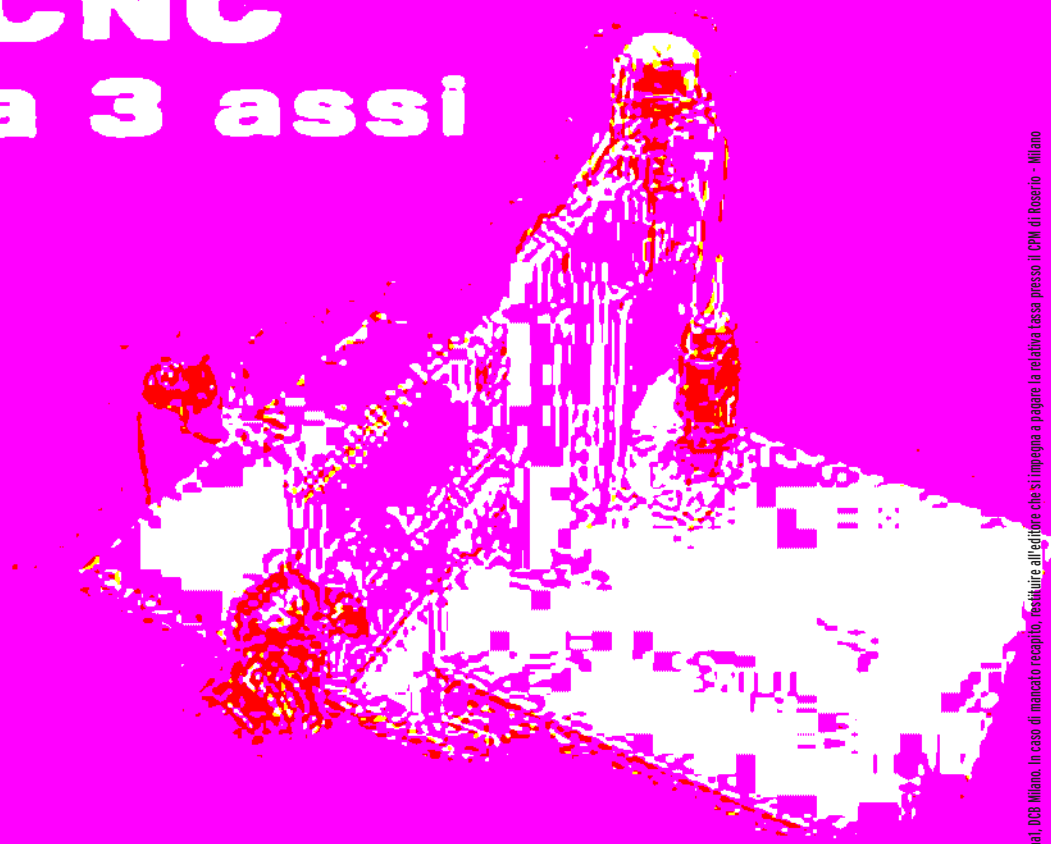
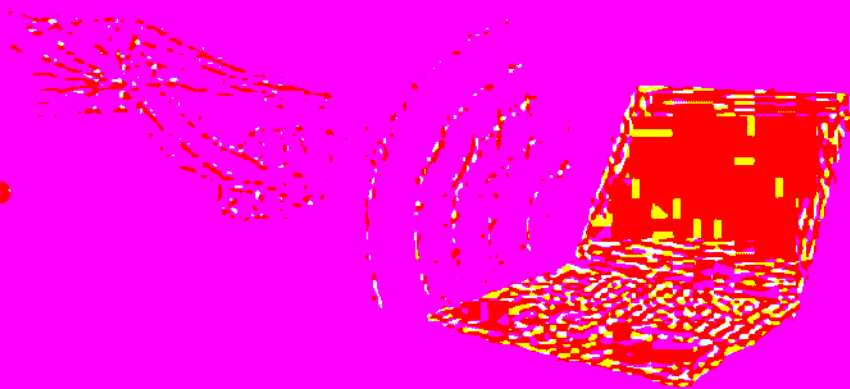
VITAMINA C
Ottimizzazione del codice

Accensione automatica del PC

L'amplificatore operazionale

€ 5,50

Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004 n. 46) art. 1, comma1, DCB Milano. In caso di mancato recapito, restituire all'editore che si impegna a pagare la relativa tassa presso il CPM di Roserio - Milano

Note dalla redazione

ENERGIA ALTERNATIVA? SI, GRAZIE!

Quando parliamo di elettronica, in tutte le sue forme, è sempre dato per scontato che sia disponibile una fonte di energia mediante la quale alimentare i dispositivi, gli apparati e tutto ciò che il concetto di "elettronica" può comprendere.
Ma è davvero così scontato?
Il problema energetico incombe minaccioso su tutti i paesi industrializzati ed è in attesa di una soluzione alla quale stanno febbrilmente lavorando le grandi Nazioni. A testimonianza di questo lavoro, l'obbligo dell'incremento della produzione energetica con fonti rinnovabili dello 0,35% all'anno, già a partire dal 2005, stabilito dal protocollo di Kyoto, nonché gli investimenti previsti dai Governi (nei prossimi due anni l'Italia investirà oltre 400 milioni di Euro nel settore dell'efficienza energetica e delle energie alternative).
Le fonti di energia alternative hanno questo nome proprio perché sono complementari e non sostitutive a quelle tradizionali ed il loro utilizzo esclusivo non coprirebbe il fabbisogno energetico di un Paese anche di medie dimensioni come l'Italia. D'altro canto le fonti alternative contribuiscono notevolmente al raggiungimento di due obiettivi: ridurre l'inquinamento e diminuire la dipendenza di un Paese da materie prime energetiche (come gli idrocarburi) sempre meno disponibili e sempre più costose.
Noi di Fare Elettronica abbiamo voluto contribuire, nel nostro piccolo, alla sensibilizzazione verso questo problema proponendo il nostro generatore eolico il cui progetto si chiude proprio con questo numero.
Per definizione di fonte alternativa, il vento non sostituirà certo il vostro contatore ENEL, ma il progetto del generatore eolico chiarirà, sicuramente, come sia possibile estrarre energia dalla forza della natura e nella maniera più indolore possibile per il pianeta.

Come avrete senz'altro visto dalla copertina, oltre all'ultima parte del progetto del generatore eolico questo numero contiene una realizzazione un po' insolita per una rivista come la nostra, ma che non abbiamo potuto fare a meno di pubblicare viste le numerose richieste giunte in redazione: *una semplice ed economica CNC a 3 assi*, la cui costruzione è davvero alla portata di tutti.
Questo non è però l'unico progetto, infatti, *l'accensione automatica del PC* desterà sicuramente l'interesse di tutti gli amanti del P2P che vogliono scaricare file durante le ore notturne quando la banda a disposizione è maggiore.
Con lo speciale di questo mese abbiamo voluto portare i lettori più giovani a scoprire *come nasce un circuito integrato*, un articolo davvero interessante ed istruttivo che illustra come sia possibile ottenere strutture micrometriche su silicio.
Con questo numero si conclude il ciclo dedicato alla *Intelligenza Artificiale*, continuano invece, con nuove ed interessanti puntate: *Elettronicando*, *Alimentatori Switching*, *Vitamina C*, *Amplificatori operazionali dalla A alla Z* e naturalmente *Mikrobasic per PICmicro*.

Concludo ricordandovi che la *Collection 2005* rimarrà in edicola fino a fine Settembre ed a giudicare dalle vendite non vi rimane molto tempo per acquistarla!
Vi auguro una piacevole lettura e vi rinnovo l'appuntamento in Edicola ad Ottobre con un numero davvero speciale che conterrà un utilissimo CD: *PROTEUS - il famoso CAE/CAD per la prima volta in Italiano*. Prenotatene una copia dal vostro edicolante!

Tiziano Galizia
t.galizia@farelettronica.com

www.farelettronica.com

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE DI REDAZIONE
Tiziano Galizia

DIRETTORE TECNICO
Maurizio Del Corso

HANNO COLLABORATO IN QUESTO NUMERO
Giovanni Di Maria, Riccardo Nicoletti, Antonio Cecere, Antonio Di Stefano, Romano Bernarducci, Maurizio Del Corso, Nico Grilloni, Marco Fabbri, Lucio Del Veneziano.

DIREZIONE • REDAZIONE • PUBBLICITÁ
INWARE srl - Via Cadorna, 27/31 - 20032 Cormano (MI)
Tel. 02.66504794 - 02.66504755 - Fax 02.66508225
info@inware.it - www.inwaredizioni.it
Redazione: redazione@farelettronica.com

PROGETTO GRAFICO E IMPAGINAZIONE
Graficonsult - Milano

STAMPA
ROTO 2000 - Via L. da Vinci, 18/20 - 20080, Casarile (MI)

DISTRIBUZIONE
Parrini & C. S.p.a. - Viale Forlanini, 23 - 20134, Milano

UFFICIO ABBONAMENTI
PARRINI & C. S.p.a. - Servizio abbonamenti
Viale Forlanini, 23 - 20134 Milano
Per informazioni, sottoscrizione o rinnovo dell'abbonamento:
abbonamenti@farelettronica.com
Tel. 02.66504794 - Fax. 02.66508225
L'ufficio abbonamenti è disponibile telefonicamente
dal lunedì al venerdì dalle 14,30 alle 17,30

Poste Italiane S.p.a. - Spedizione in abbonamento Postale
D.L. 353/2003 (conv. In L. 27/02/2004 n. 46) art. 1, comma1, DCB Milano.
Abbonamento per l'Italia: € 45,00
Abbonamento per l'estero: € 115,00

Gli arretrati potranno essere richiesti, per iscritto, al seguente costo:
Numero singolo: € 7,50
Numero doppio: € 9,00
Numero con allegato: € 8,50

Autorizzazione alla pubblicazione del Tribunale di Milano n. 647 del 17/11/2003.
Iscrizione al R.O.C. n. 11035 19/11/2003

RICHIESTE DI ASSISTENZA
Per richiedere assistenza o chiarimenti sugli articoli pubblicati, vi preghiamo di contattare direttamente l'autore. Se questo non fosse possibile, utilizzate il modulo di contatto che trovate sul nostro sito web www.farelettronica.com.

COLLABORARE CON FARE ELETTRONICA
Le richieste di collaborazione vanno indirizzate all'attenzione di Tiziano Galizia *(t.galizia@farelettronica.com)* e accompagnate, se possibile, da una breve descrizione delle vostre competenze tecniche e/o editoriali, oltre che da un elenco degli argomenti e/o progetti che desiderate proporre.

ELENCO INSERZIONISTI

Pratica

pag. 90

Teoria

pag. 56

pag. 12

Risorse

Speciale

Sommario

243
Settembre
2005

Scrivete a:
MAILBOX
REDAZIONE DI
FARE ELETTRONICA
Inware s.r.l.
Via Cadorna, 27/31
20032, Cormano (MI)

Oppure inviate un'email a:
mailbox@farelettronica.com

Mailbox

*Questa rubrica ospita le richieste più interessanti pervenute dai lettori.
Per quanto possibile verrà data risposta a tutte le richieste pervenute via email.*

GENERATORE DI TENSIONE 1-10V

Ho l'esigenza di costruire un'interfaccia che riesca, al ricevimento di un impulso dato da un pulsante, a far uscire una tensione da 1 a 10 volt ad ogni impulso ricevuto. Più semplicemente ad ogni impulso la tensione salirà di 1 volt fino al massimo di 10V. Il tutto mi servirebbe per simulare nel mio acquario alba e tramonto pilotando poi un trasformatore HF che permette di variare la luminosità di un tubo al neon.
Il Trasformatore HF dispone di un ingresso da 1 a 10 volt per far variare la luminosità da 3% a 100%.

Riccardo Tomasetti

Il modo più semplice è quello di utilizzare un contatore decimale e collegare la sua uscita ad

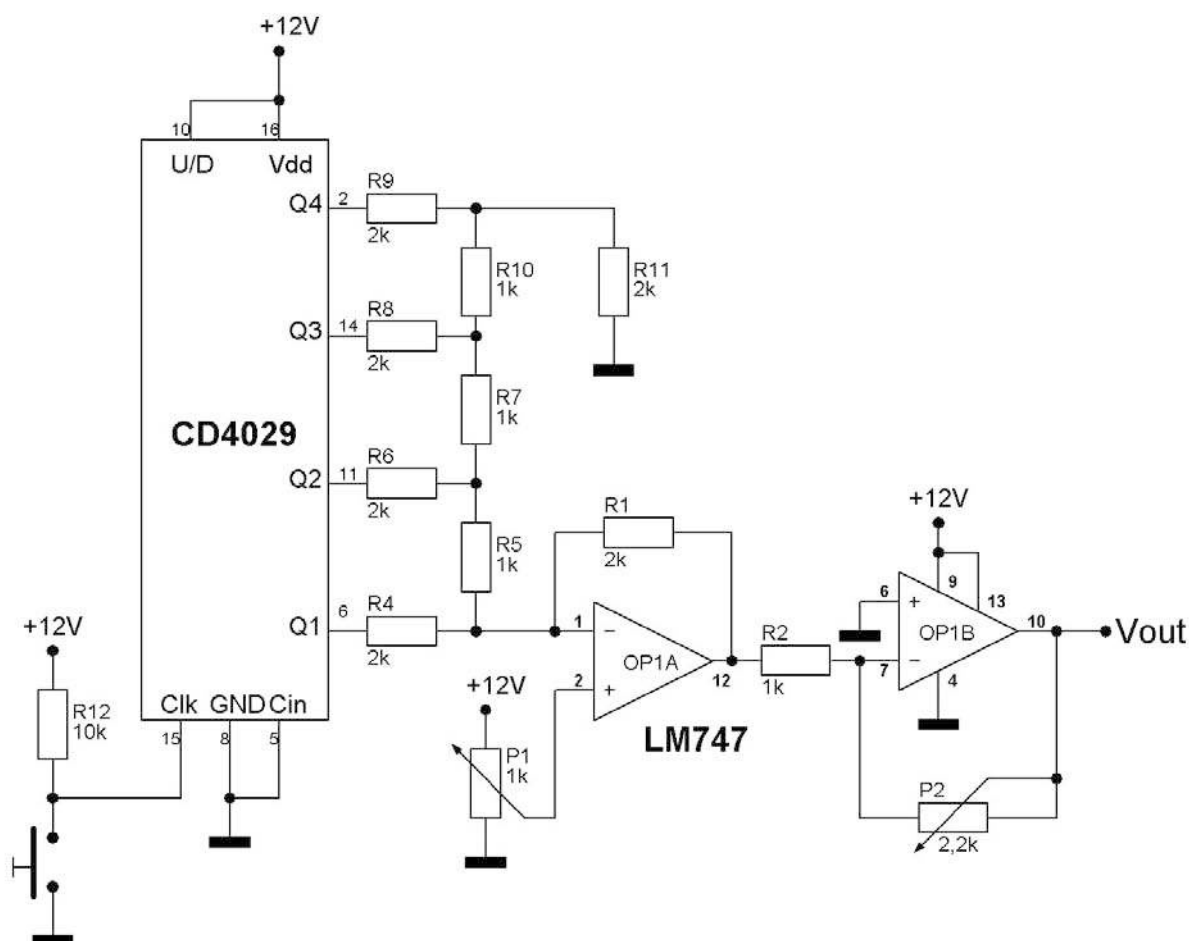



Figura 1 *Il generatore di gradinata*

"Richieste, chiarimenti, dubbi e commenti dai lettori"

un convertitore AD realizzato ad esempio con un amplificatore operazionale. Lo schema è quello di figura 1. Ad ogni pressione del pulsante il contatore incrementa il conteggio quindi il convertitore trasforma il risultato in una tensione continua. Il trimmer P2 consente di regolare il guadagno in modo che al massimo valore di uscita del contatore (9), corrisponda una tensione di 10V.
Il trimmer P1 permette invece di avere all'ingresso del convertitore una tensione continua in modo da ottenere 1V in uscita quando il contatore è all'inizio del conteggio. Per eliminare i rimbalzi del pulsante è possibile inserire un latch antirimbalzo all'ingresso Clk del contatore.

ANTENNA AM/FM ATTIVA

Vorrei costruire un'antenna attiva per la ricezione in AM/FM ed SW. Potete pubblicare uno schema semplice a tal proposito?

Giancarlo Givoli

In figura 2 lo schema richiesto. Per L1 si deve scegliere 470µH per l'uso in AM ed FM, mentre per le onde corte SW si dovrà usare una induttanza da 20µH. L'alimentazione può essere ricavata da una pila alcalina. L'antenna usata nello schema è una comune antenna a stilo da 18" telescopica, ma vanno altrettanto bene anche un comune spezzone di cavo di rame o una barra metallica della stessa lunghezza. L'elemento attivo è un JFET a canale N normalmente usato come amplificatore UHF/VHF e può essere sostituito con un NTE451.

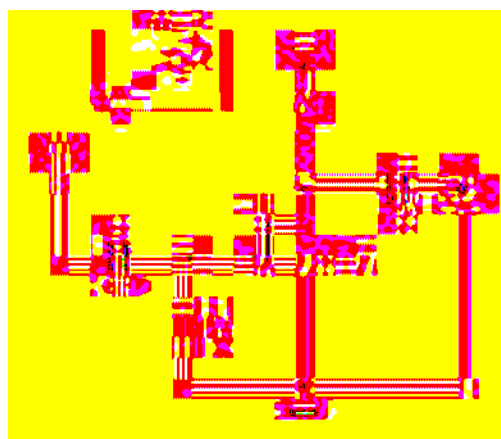
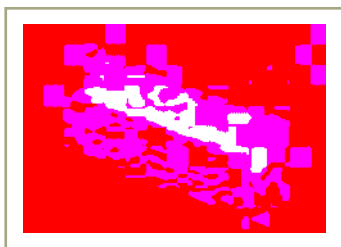



Figura 2 Antenna attiva AM/FM/SW

Rubrica di notizie e novità dal mondo dell'elettronica.

Prima pagina

243-01 CONNETTORI “HARD METRIC”: COM-MET 2HM

Compel Electronics SpA, annuncia l’ampliamento del suo catalogo di connettori metrici. In aggiunta alle esistenti serie COM-MET 2.5 e COM-MET 2 (IEC 61076-4-104) e per soddisfare la richiesta dei propri clienti, che oggi utilizzano diversi standard di connettori metrici da scheda, Compel ha sviluppato la propria famiglia di connettori metrici a passo 2 mm secondo la normativa IEC 61076-4-101, altrimenti noti come “Hard Metric”. Questa famiglia, denominata COM-MET 2HM, prevede connettori maschi dei tipi A – AB – B – B22 – C – D – E – F, con terminazioni Press-fit P2 – R1 – R2 – R3.

243-02 ADC BANDA LARGA A 125 MSPS, 14 BIT A BASSO CONSUMO

Linear Technology Corporation annuncia il nuovo LTC2255, un convertitore analogico/digitale (ADC) a 125 Msps e 14 bit con eccellenti caratteristiche prestazionali e ridotto consumo energetico. Rispetto al prodotto direttamente concorrente a 14 bit, l'LTC2255 consuma il 49% di energia in meno a soli 395mW, riducendo sensibilmente il budget dei consumi e i requisiti termici dei dispositivi a più canali. Ciò garantisce un vantaggio significativo per le applicazioni che richiedono livelli elevati di efficienza e raffreddamento, come ricevitori satellitari, stazioni base wireless e dispositivi elettronici portatili.
Appartenente ad una famiglia di prodotti con estesa compatibilità a livello di pin, l'LTC2255 è offerto in un package QFN di 5 x 5 mm, estremamente comodo date le dimensioni ridotte, con condensatori bypass integrati, e richiede un numero limitato di piccoli componenti esterni. L'LTC2255 elimina la necessità di utilizzare condensatori di disaccoppiamento grandi e costosi, offrendo la migliore soluzione in piccole dimensioni, in grado di risolvere i problemi di spazio sulla scheda e offrire progettazioni più compatte ed economicamente convenienti.

L’LTC2255 consente anche di soddisfare le esigenze delle ultime tecnologie 3G e 4G, delle applicazioni WiMAX e di altre applicazioni wireless a banda larga, in cui gli ADC ad alte prestazioni giocano un ruolo fondamentale nella gestione delle esigenze di un traffico di rete sempre più intenso. Per i progettisti di sistemi per stazioni base wireless la riduzione dei consumi è un fattore importante per tagliare i costi operativi generali di sistema. Inoltre, grazie alla combinazione di elevata velocità di campionamento, bassa corrente e risoluzione a 14 bit, il nuovo convertitore è la soluzione ideale per le apparecchiature di test e strumentazione a batteria e alte prestazioni.
L'LTC2255 offre ottime prestazioni del segnale di ingresso a basso livello grazie all'elevata linearità ed è progettato con un buon margine per quanto riguarda la velocità di campionamento allo scopo di garantire prestazioni affidabili su un ampio intervallo di temperature. Alla velocità di campionamento di 125Msps, le prestazioni CA offerte sono ottime, con 72,1 dB SNR e 85 dB SFDR a 70 MHz.

PER SAPERNE DI PIÙ

Per approfondire le notizie riportate in questa rubrica, visitate il sito **www.farelettronica.com/primapagina**

In alternativa potete scrivere a:

Inware Edizioni
Servizio Prima Pagina
Via Cadorna 27 - 20032 Cormano (MI)

Indicando il codice riportato accanto al titolo della notizia (esempio 243-02).

243-03 OMNIVISION TECHNOLOGIES LANCIA LA TECNOLOGIA INNOVATIVA OMNIPIXEL™

OmniVision Technologies Inc., costruttore indipendente di Sensori CMOS CameraChip™ per applicazioni video, ha annunciato la nuova Tecnologia Innovativa OmniPixel™ per applicazioni avanzate nell'ambito dei sensori d'immagine.
Questa tecnologia permette ai Sensori CMOS di futura generazione di avere una sensitività, una risoluzione, una fedeltà dei colori ed un basso rumore simili ai sensori CCD, mantenendo tutti i vantaggi quali il basso costo, l'alta integrazione, il basso consumo, le ridotte dimensioni, l'ampio range dinamico e la possibilità di deviare tra still-image e video capture.

Altre caratteristiche salienti quali l'auto-focus, lo zooming, il panning ed il controllo meccanico dello shutter danno a questi sensori CMOS un ulteriore vantaggio prestazionale.
La risoluzione di questi sensori, coprirà un intervallo molto ampio:

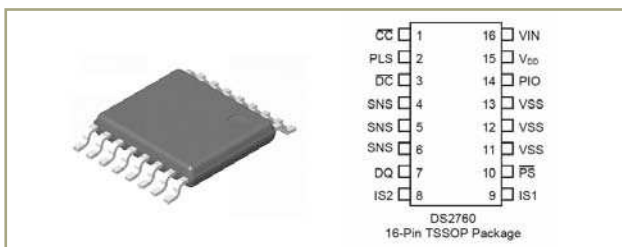
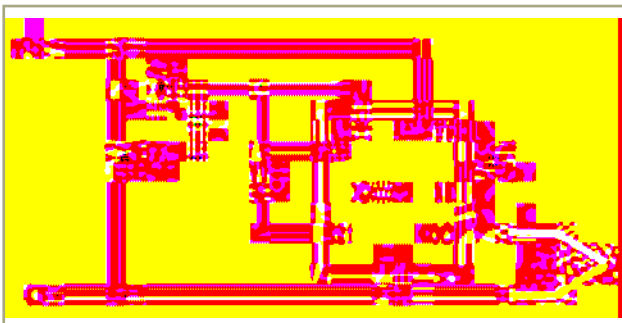
- PAL e NTSC (Analogici).
- VGA, 1.3 MPixel, 2 MPixel, 3.2 MPixel e 5 MPixel (Digitali).

Dal blocco note di Fare Elettronica una raccolta di idee da tenere sempre a portata di mano.

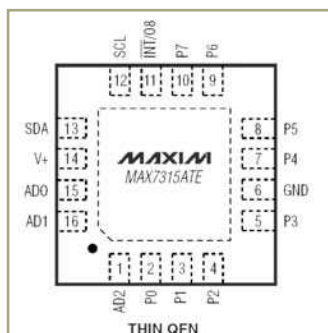
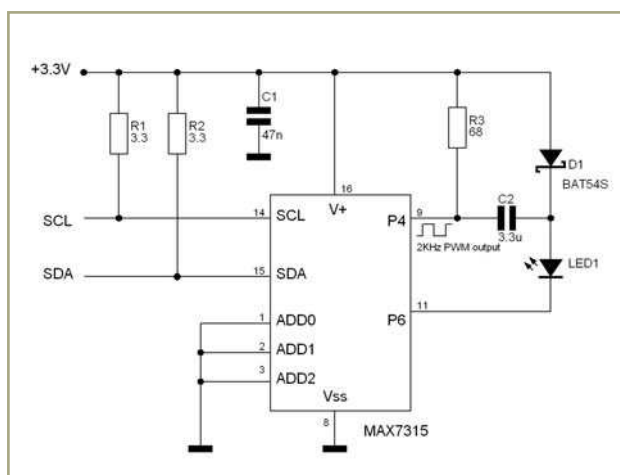
Notepad

Questa rubrica ha lo scopo di fornire degli schemi applicativi o idee di progetto dei componenti elettronici più interessanti, selezionati per voi dalla redazione. Tutti gli schemi presentati sono elaborazioni di quelli ufficiali proposti dai produttori nella documentazione ufficiale.

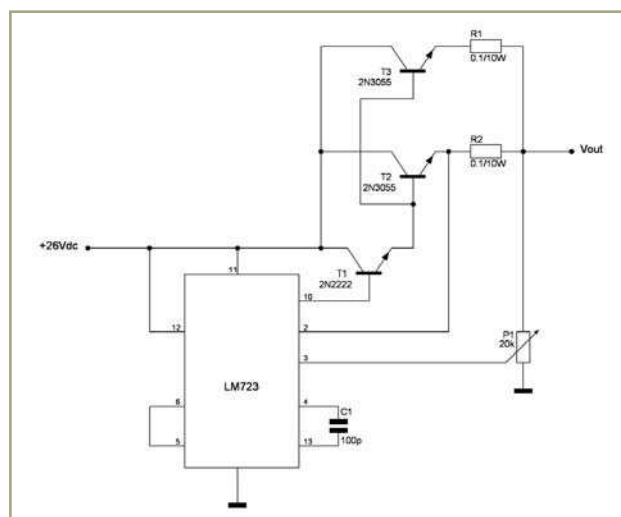
TERMOCOPPIA 1-WIRE

Utilizzando il chip DS2760 (un controllore per batterie agli ioni di Litio) è possibile trasformare una normale termocoppia standard di tipo E in una termocoppia 1-Wire.
La coppia D1, C1 funziona da raddrizzatore ad un semionda e permette di ottenere l'alimentazione del DS2760 durante la trasmissione dati.

Il DS2760 contiene un convertitore AD a 10 bit ed una EEPROM di 32bytes in cui è possibile memorizzare informazioni personalizzate per il sensore.

UTILIZZARE UN GPIO EXPANDER I2C COME POMPA DI CARICA

Con il circuito di figura è possibile trasformare un I2C GPIO Expander MAX7315 in una pompa di carica.

Un utilizzo tipico di questo circuito è il pilotaggio di LED blu con una tensione di 3,3V. La tecnica usata è quella di realizzare un raddrizzatore D1, C2 su una delle uscite (P4 in figura) e pilotare questa in PWM.
Quando l'uscita è bassa C2 si carica alla tensione di alimentazione, mentre quando l'uscita è alta la tensione sul condensatore si somma a quella di alimentazione rendendo possibile l'accensione del LED.
Con una tensione di alimentazione di 3,3V, il circuito è in grado di erogare circa 15mA, a 3,0V può erogare 10mA mentre a 2,0V la massima corrente erogabile scende a circa 3mA.

ALIMENTATORE DA 10A

Lo schema di figura mostra come ottenere un alimentatore variabile da 3V a 18V con una corrente di uscita massima di 10Amp. Tutti i transistori 2N3055 dovranno essere montati su aletta di raffreddamento e se ne possono aggiungere altri, connessi come T3, per ottenere correnti maggiori. Il circuito è molto stabile, infatti prove di laboratorio hanno dimostrato che a 10V di uscita la tensione di uscita si abbassa di appena 0,09V passando dal funzionamento a vuoto ad un carico da 10Amp. Il cuore del circuito è un LM723, un regolatore di tensione particolarmente adatto alla realizzazione di alimentatori lineari.

Le so tutte!!!

Risposta al quesito codice LST24106

Il circuito è composto da una rete di transistori pmos che collega l'uscita Y a +Vdd ed una rete di nmos che collega la stessa uscita alla massa del circuito. Sapendo che un pmos conduce solo quando il suo ingresso è a livello basso ed un nmos conduce quando l'ingresso è a livello alto, si deduce che l'uscita Y è a livello alto quando T1 e T3 conducono oppure quando T4 conduce. T1 e T3 conducono quando sia A che B sono a livello basso mentre T4 conduce quando C è a livello basso. Risulta quindi $Y=(\bar{A}$ and $\bar{B})$ or $\bar{C}$ (dove con $\bar{A}$, $\bar{B}$ e $\bar{C}$ si intendono le negazioni di A, B e C rispettivamente). Si noti che la rete di nmos consente di non avere ambiguità in quanto quando T1 e T3 conducono, T2 e T5 sono interdetti e analogamente quando T4 conduce T6 è interdetto.

Il vincitore di GIUGNO 2005 (Vincitore LST24005)
I nostri complimenti a Belvisi Agostino di Borgo Piave (LT)
che vince un abbonamento a Fare Elettronica!

Elettronicando

Ci occuperemo in questa puntata dell'Amplificatore Operazionale, uno dei componenti più diffusi in elettronica. Le sue applicazioni sono svariate: dagli utilizzi in elettronica analogica ai generatori di forma d'onda, ai comparatori, ai circuiti per elaborazione e conversione di segnali, agli strumenti elettronici. Faremo una panoramica dei componenti più diffusi ed introdurremo le configurazioni di amplificatore invertente e non–invertente, simulandole con Cadlogix.

L'AMPLIFICATORE OPERAZIONALE

Un po' di storia

Il termine "operazionale" deriva dal fatto che, in origine, tale componente era usato nei calcolatori analogici per svolgere operazioni matematiche su segnali elettrici. Negli anni '40 i primi esemplari vennero realizzati con tubi a vuoto, mentre con l'avvento del transistor bipolare furono costruiti i primi esemplari a componenti discreti.

Il primo amplificatore operazionale monolitico fu il µA702 e fu progettato nel 1965 da Bob Widlar, allora progettista presso la Fairchild. Al µA702 seguì il µA709. Ebbe molta popolarità sul momento, ma poi denunciò una serie di problemi. Quando uscì sul mercato costava più di 100 dollari. Widlar andò alla National, dove progettò l'LM301, non dotandolo però di compensazione in frequenza interna.

Nel frattempo la Fairchild studiò la risposta a questo componente, che sarebbe arrivata con il famoso 741, che possedeva i vantaggi del 301 ed aveva in più la compensazione interna. Questo gli consentiva di conservare due pin per la regolazione dell'offset. Poiché molti circuiti non richiedono la regolazione dell'offset, il 741 in usi normali non richiede altri componenti che la sola rete di reazione. Il resto è storia! Il 741 divenne lo standard industriale per gli amplificatori operazionali per molti anni.

Adesso esistono numerosi componenti simili al 741, ma con alcuni vantaggi in più, come l'ingresso a FET, con due o quattro unità, versioni con caratteristiche migliorate, compensati e non compensati, eccetera.

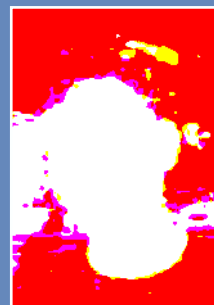
Il simbolo grafico

L'amplificatore operazionale è un tipico esempio di *amplificatore retroazionato*, in quanto non viene mai usato in assenza di reazione. Le sue caratteristiche, uniche ed eccezionali, ne hanno determinato una vastissima diffusione. Esistono infatti centinaia di differenti A.O. (oppure "O.A." o anche "Op-Amp", come lo chiamano gli americani), ciascuno con caratteristiche particolari.

Figura 1 *Il simbolo dell'A.O.*

L'A.O. è rappresentato col noto triangolo, nel quale figurano due ingressi ed un terminale di uscita (figura 1).

L'amplificatore operazionale possiede due ingressi, un'uscita e due terminali per l'alimentazione; l'alimentazione è di solito *bilanciata* (±15 V) e questo consente all'uscita di fornire tensioni comprese all'interno di questo

L'amplificatore operazionale

di Riccardo Nicoletti
r.nicoletti@farelettronica.com

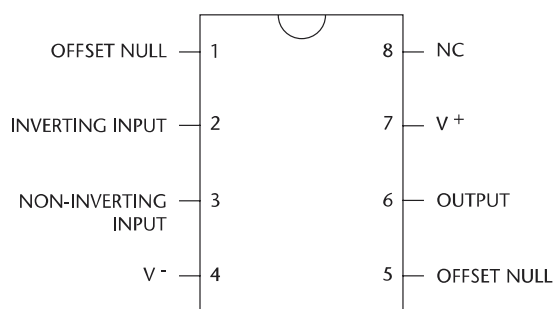



Figura 2 *Piedinatura dell'A.O. LM741 della National Semiconductor in package 8 pin DIL*

intervallo. Spesso negli schemi elettrici i due terminali di $+V_{cc}$ e $-V_{cc}$ non figurano.
I due terminali di ingresso sono contrassegnati da un "+" e da un "-"; il primo si chiama "*ingresso non invertente*" mentre il secondo è "*l'ingresso invertente*". I simboli "+" e "-" non significano che il primo terminale deve sempre trovarsi ad un potenziale superiore rispetto al secondo, ma stanno a significare che l'uscita diventa positiva quando l'ingresso "+" si trova ad un potenziale superiore rispetto al "-", mentre va negativa quando accade il contrario.
Il simbolo grafico può anche riportare i due pin per la regolazione dell'offset, denominati "offset null" oppure "balance" (figura 1).

Il padre degli Op-Amp: il 741

Il padre di tutti gli amplificatori operazionali fu il "741" della National Semiconductor. Sebbene oggi molti suoi successori lo surclassino per velocità, guadagno, basso rumore, eccetera, esso lavora ancora bene ed è un riferimento per la sua semplicità d'uso.
La sua risposta in frequenza è compensata, e quindi il componente non fa brutti scherzi durante il funzionamento.

Figura 4 *Contenitore metallico di tipo SO-5*

Abita di solito in un package 8-pin di tipo DIL; per il suo successo duraturo molti suoi successori hanno mantenuto la stessa piedinatura per conservare la compatibilità dei collegamenti. Si trova in diverse versioni (LM741, NE741, µA741), e ne esistono alcuni modelli seguiti da una lettera, es. LM741C, poiché qualche costruttore ha apportato qualche miglioria. Si può anche trovare in contenitore metallico SO-5 (figura 4), la cui piedinatura è riportata in figura 5.
È inoltre disponibile in contenitore Flatpack, la cui piedinatura è riportata in figura 6.
Possiede una resistenza di ingresso di circa 2 MΩ, una corrente di bias in ingresso di 80 nA, uno slew-rate di 0.5 V/µs ed un guadagno in bassa frequenza di circa 200000.

Figura 3 *Piedinatura dell'A.O. LM741 della National Semiconductor in package ceramico DIL*

Figura 5 *Piedinatura LM741 per contenitore metallico SO-5*

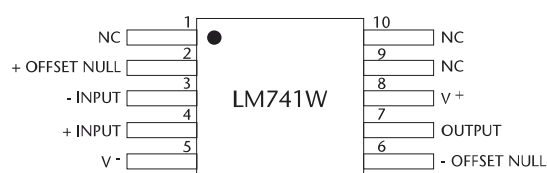



Figura 6 *Contenitore ceramico Flatpack per LM741*

A causa della sua compensazione in frequenza il suo guadagno decresce rapidamente; vale 1000 a 1 kHz, 100 a 10 kHz ed è circa unitario a 1 MHz (figura 7).
Per la versione µA741, la Texas Instruments dichiara i valori dei parametri riportati nella figura 8. I valori riportati vi saranno chiarissimi dopo aver studiato lo speciale di Nico Grilloni, che ci ha illuminati su come interpretare quanto dichiarato nei datasheet degli operazionali. La Texas Instruments conserva la piedinatura della National per garantire la compatibilità dei due integrati (figura 9).
Il datasheet riporta anche delle tabelle molto interessanti, dove sono elencati i contenitori da utilizzare in funzione della temperatura di utilizzo

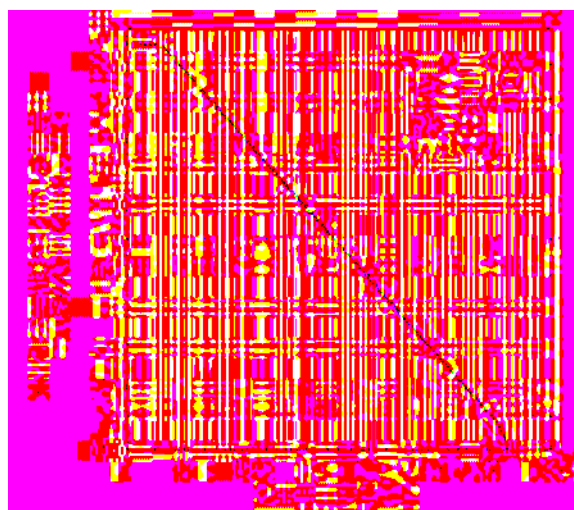



Figura 7 *Guadagno ad anello aperto in funzione della frequenza per 741 (Texas instruments)*

(figura 10) o a seconda della potenza dissipabile (figura 11).

Un altro esempio: l'amplificatore LF411

Tra i tanti esempi di A.O. presenti in commercio,

Figura 8 *Dati elettrici per µA741 (Texas Instruments).*

OFFSET N1 1 — 8 NC
IN - 2 — 7 V_{CC} +
IN + 3 — 6 OUT
V_{CC} - 4 — 5 OFFSET N2

Figura 9 Piedinatura µA741 in package 8-pin DIL compatibile National Semiconductor

un personaggio quasi storico si chiama LF411 (detto semplicemente "411" per gli addetti ai lavori), introdotto in origine dalla *National Semiconductor*.
Esso si trova incapsulato nel package mini-DIP (dual in line package), come riportato in figura 12. La sua piedinatura è quella di figura 13.
Questo componente è molto economico e facile da usare. Si trova anche in una versione migliorata (LF411A) ed anche in un contenitore che ne contiene due (LF412, chiamato "*dual*" op-amp). Questo componente può essere un buon punto di partenza per i vostri esperimenti.
La piedinatura del chip LF411 è riportata in figura 13 (ricordatevi di contare sempre i piedini in verso ANTI-orario, partendo dal riferimento che trovate sul contenitore). Nico Grilloni ci ha già spiegato come utilizzare i pin denominati "*offset null*" o anche "*balance*" (FE 237, pag. 45).
Dentro il chip LF411 c'è un pezzo di Silicio sul quale sono realizzati 24 transistori (di cui 21 BJT e 3 FET) 11 resistori ed 1 condensatore. Mi sovviene una domanda: "ma cosa ci fanno dei FET dentro l'operazionale?".

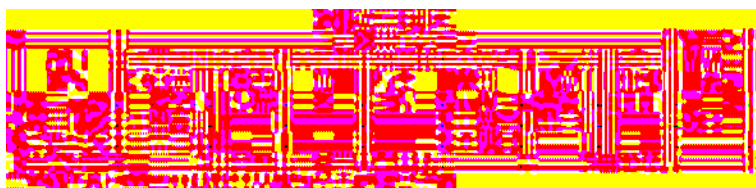



Figura 10 Package disponibili in funzione della temperatura di utilizzo (Texas Instruments)

Figura 11 Package e potenze dissipabili in relazione alla temperatura (Texas Instruments)

Le varie tipologie di componenti

Gli amplificatori operazionali vengono realizzati fondamentalmente con due tecnologie: bipolari e a FET.
Sono molto diffusi gli amplificatori che utilizzano i FET nel loro stadio di ingresso e poi sono realizzati con BJT (figura 15). I FET in ingresso conferiscono una elevata impedenza di ingresso, che costituisce una caratteristica peculiare dell'A.O.
Tali componenti vengono indicati con la dicitura "*JFET Input* Operational Amplifier" (es. LF411).
L'amplificatore operazionale può essere schematizzato in forma semplificata, come composto da tre stadi. Il primo è l'amplificatore differenziale di ingresso (a FET, per garantire un'elevata resistenza di ingresso, cerchiato in blu in figura 15), il secondo è uno stadio amplificatore che realizza anche la compensazione in frequenza (evidenziato in rosso, figura 15), mentre lo stadio finale realizza anche una adeguata capacità di pilotaggio in uscita (in verde in figura 15).
Alcune case costruttrici realizzano circuiti integrati contenenti più di un amplificatore al loro interno. Solitamente se ne trovano con due o quattro, denominati rispettivamente "*Dual*" e "*Quad*". Quindi il componente LF412, denominato "*Dual JFET Input Operational Amplifier*", è un componente che contiene due amplificatori operazionali con ingresso a JFET (figura 14), mentre l'LM324 ne contiene quattro (figura 16). I nomi dei componenti disponibili in commercio contengono le caratteristiche speciali di quel particolare operazionale, perciò leggetele sempre con attenzione!
Ad esempio, l'integrato LF412 di cui parlavamo prima, viene chiamato "*Low Offset, Low Drift Dual JFET Input Operational Amplifier*". Analizzando il titolo si vede che l'ultima parte dice che tipo di componente è l'LF412, cioè un

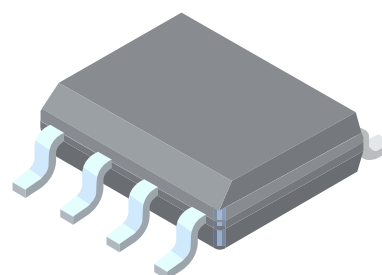



Figura 12 Il package dell'A.O. LF411 della National Semiconductor

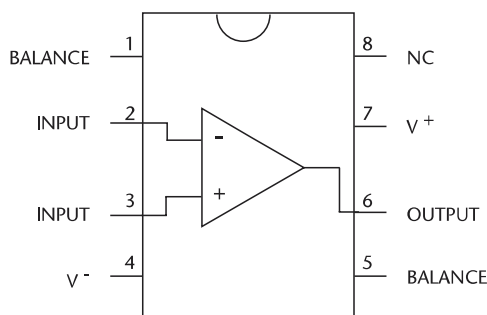



Figura 13 La piedinatura dell'A.O. LF411 della National Semiconductor

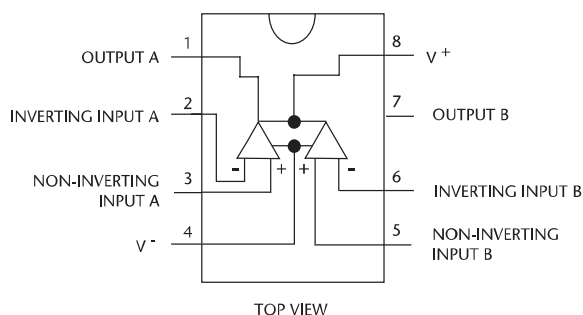



Figura 14 Piedinatura LF412

doppio amplificatore operazionale, mentre la prima parte evidenzia le seguenti caratteristiche speciali: *bassa tensione di offset, basso drift ed ingresso a JFET.*
Se avete studiato il bellissimo speciale realizzato da Nico Grilloni capirete benissimo le specifiche di questo componente.
Vediamo altri esempi:

LF155/LF256/LF257/LF355/LF357 – *JFET Input Operational Amplifiers*, (National Semiconductor) – particolarmente indicati per applicazioni dove si richiede basso rumore, basso drift e bassa corrente di bias. Possiedono uno slew-rate tipico di 12 V/µs (156, 256, 356) e di 50 V/µs (257, 357). Particolarmente indicati per convertitori D/A e A/D veloci, integratori di precisione ad alta velocità, circuiti S&H.

LM124/LM224/LM324/LM2902 – *Low Power Quad Operational amplifiers,* (National Semiconductor) – contiene quattro A.O. indipen-

denti, con alimentazione singola (V_{cc} =15 V) o doppia.

LM148JAN – *Quad 741 Op Amps*, (National Semiconductor) – contiene quattro A.O. di tipo 741 indipendenti.
Le correnti di offset e di bias sono più piccole di quelle di un 741 originale. Piedinatura compatibile con LM324 (figura 16).

Figura 15 Schema semplificato della struttura di un A.O. con ingresso a FET (LF411)

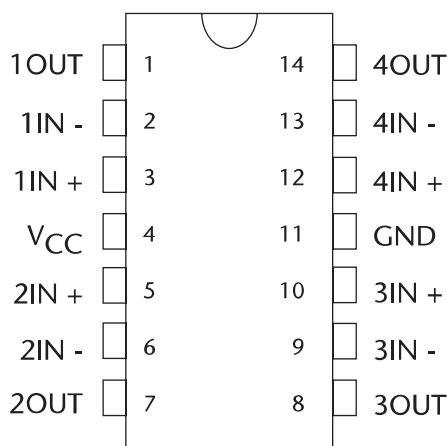



Figura 16 Piedinatura del LM324, che contiene quattro A.O.

LMC8101 – *Rail-to-Rail Input and Output*, 2.7V *Op Amp in micro SMD package with Shutdown*, (National Semiconductor) – Ecco un bel nome completo! Questo è un A.O. di tipo CMOS, ideale per applicazioni *Rail-to-Rail* per basse tensioni. La dicitura *Rail-to-Rail* indica la capacità della tensione di uscita (o di ingresso) di raggiungere valori estremamente vicini a quelli dell'alimentazione.
Il componente in esame può essere alimentato da 2.7V a 10V. L'integrato LMC8101 è contenuto in un package SMD ad 8-pin, che occupa il 75% in meno di spazio rispetto ad un contenitore SOT23-5.
Possiede anche la possibilità di essere spento (modalità "shutdown") e di assorbire in tal caso una corrente di alimentazione di solo 1μA. È l'ideale per circuiti alimentati a batteria, apparati wireless e applicazioni industriali.

LE CARATTERISTICHE DELL'A.O.

Le caratteristiche del componente ideale

L'amplificatore operazionale ideale possiede (o "dovrebbe" possedere) alcune caratteristiche del tutto particolari, e cioè: impedenza di ingresso infinita, impedenza di uscita nulla, guadagno di tensione infinito, guadagno di tensione di modo comune nullo, tensione di uscita nulla quando gli ingresso sono allo stesso potenziale, tensione di uscita variabile istantaneamente (slew rate infinito).

Queste proprietà dovrebbero poi essere indipendenti dalla temperatura e dalle variazioni della tensione di alimentazione.
Nella realtà queste caratteristiche non sono

Proprietà	Caso ideale	Nella pratica
Guadagno di tensione ad anello aperto	Infinito	Molto alto (≥10^4)
Rapporto di Reiezione di modo comune	Infinito	Alto (≥70 dB)
Impedenza di ingresso	Infinita	Alta (≥10 MΩ)
Impedenza di uscita	Nulla	Bassa (<500Ω)
Corrente di ingresso	Nulla	Bassa (<0.5 μA)
Tensioni e correnti di offset	Nulle	Basse (<10 mV, <0.2 nA)

Figura 17 Confronto tra parametri ideali e reali degli A.O.

Figura 18 *Schema di principio di un amplificatore invertente con A.O.*

Figura 19 *Circuito completo per amplificatore invertente con A.O.*

rispettate, anche se alcune si avvicinano molto al caso reale.
La tabella di figura 17 riporta un confronto tra quello che *vorremmo* avere e quello che *abbiamo* in realtà.
Analizzando i dati riportati in figura 8, si vede come i dati reali si discostano dai dati ideali, seppur in maniera contenuta.

Le regole d'oro

Ci sono due regole d'oro da non dimenticare, mentre si lavora con gli amplificatori operazionali con reazione esterna.
Prima di tutto teniamo presente che il guadagno di tensione è così elevato che una frazione di millivolt presente tra i due terminali di ingresso farà schizzare la tensione di uscita ben oltre il suo limite; questo ci permette di esporre la *regola numero uno*: la tensione di uscita reagirà sempre in modo da far sì che la differenza di tensione tra i due terminali di ingresso sia zero (attraverso la rete di reazione). In secondo luogo, gli operazionali assorbono una corrente di ingresso irrisoria (0.2 nA per l'LF411 o qualche picoampere per quelli con ingresso a FET); quindi ignoriamo questo ed enunciamo la *regola due*: la corrente che entra nei terminali di ingresso è nulla.

Alcuni componenti consigliati

Alcuni amplificatori operazionali meritano di essere citati nel panorama commerciale, per sottoporli alla vostra attenzione.

LF 411 – è un componente valido per applicazioni generiche – possiede V_{offset} (max) = 2 mV (0.5 mV per la versione LF411A) ed una deriva massima di 20 µV/°C. la corrente di ingresso massima è 0.2 nA mentre il suo slew-rate tipico è dichiarato in 15 V/µs. Piedinatura compatibile con 741.
Indicato per integratori veloci o convertitori D/A veloci.

AD711K – è la versione migliorata dell'LF411. Infatti possiede V_{offset} (max) = 0.5 mV ed una deriva massima di 10 µV/°C. la corrente di ingresso massima è 0.05 nA mentre il suo slew-rate tipico è dichiarato in 20 V/µs.

OP27E – è un componente di precisione con basso rumore. Possiede V_{offset} (max) = 0.025 mV ed una deriva massima di 0.6 µV/°C. la corrente di ingresso massima è 40 nA mentre il suo slew-rate

Figura 20 *Simulazione dell'amplificatore invertente di figura 19*

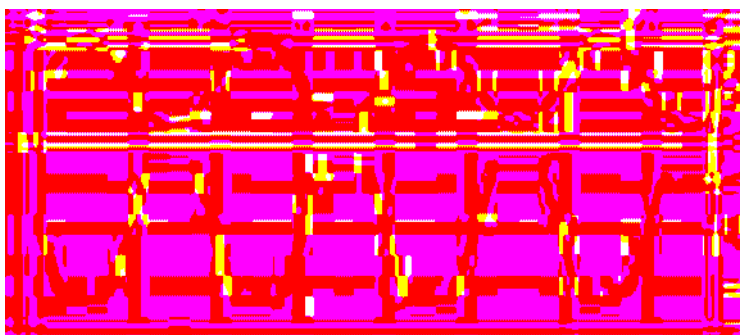



Figura 21 *Simulazione dell'amplificatore invertente di figura 19 con V_{in}(max)=2V.*

tipico è dichiarato in 2.8 V/µs.

LM308 – amplificatore di precisione, con caratteristiche di deriva garantite. Possiede V_{offset} (max) = 7.5 mV ed una deriva massima di 30 µV/°C. la corrente di bias in ingresso massima è 7 nA mentre il suo slew-rate tipico è dichiarato in 0.15 V/µs.

TL081 – componente ad alta velocità e basso costo; possiede V_{offset} (max) = 3 mV ed una deriva tipica di 10 µV/°C. la corrente di bias in ingresso massima è 0.2 nA mentre il suo slew-rate tipico è dichiarato in 13 V/µs. Piedinatura compatibile con LM741. Possiede basso rumore e bassa deriva in tensione, ma per applicazioni dove queste siano critiche è consigliato l'uso degli LF356.

LM833 – *Dual Audio Operational Amplifier* (National Semiconductor) – possiede prestazioni ottimizzate per utilizzi in pre-amplificatori di tipo audio; piedinatura compatibile con altri A.O. in chip "dual".

LMC6041 – *CMOS Single Micropower Operational Amplifier* (National Semiconductor) – ideali per sistemi che richiedono un consumo di potenza estremamente basso (sistemi di alimentazione, S&H). Possiede un valore di corrente di ingresso tipico di 2 fA!

OP77 – (Analog Devices) Possiede una tensione di offset estremamente bassa ed una ottima linearità di guadagno in frequenza. Le sue caratteristiche, unite ad un consumo di potenza di soli 50 mW, lo rendono particolarmente indicato in strumentazione elettronica di precisione.

UN PAIO DI CIRCUITI FONDAMENTALI

Amplificatore invertente

Le analisi dei circuiti contenenti amplificatori operazionali si effettuano tenendo presenti le regole d'oro enunciate in precedenza. Lo schema di principio di figura 18 riporta la configurazione di amplificatore invertente con operazionale. Il punto B è a massa e, per la regola 1 lo è anche il punto A.
Questo implica che il potenziale ai capi di R_2 è V_{out} ed il potenziale ai capi di R_1 è V_{in}. Poiché gli ingressi non assorbono corrente (regola 2) si ha

$$V_{out}/R_2 = - V_{in}/R_1$$

e quindi

$$V_{out} = -\frac{R_2}{R_1} V_{in}$$

cioè il guadagno in tensione di questa configurazione è

$$A_v = -\frac{R_2}{R_1}$$

ovvero è dipendente esclusivamente dai componenti passivi che compongono la rete di reazione, con i vantaggi che ne conseguono; il segno "-" indica che l'uscita è sfasata di 180° rispetto all'ingresso, quindi è "invertita".
Per simulare il circuito, non dimentichiamo di alimentare l'operazionale con Vcc=+15V (pin 7) e –Vcc=-15V (pin 4) (figura19).
Supponiamo R_1= 10 kΩ ed R_2=100 kΩ, quindi Av= - 10; in figura 20 è riportata la simulazione per un segnale di ingresso con V_M= 0.5 V ed f = 100 Hz; si noti lo sfasamento presente tra segnale di ingresso e di uscita.
L'amplificatore operazionale è un amplificatore "in continua", quindi anche un segnale continuo può essere amplificato; questa prova è lasciata ai Lettori. Gli amplificatori operazionali possono fornire una tensione di uscita che si avvicina al valore di Vcc (modalità "rail-to-rail"); l'esempio di figura 19 è simulato con Vin(max)=0.5 V, quindi Vout(max)= 5 V e tutto funziona bene, poiché il circuito è alimentato con +Vcc=15V e –Vcc=-15 V.

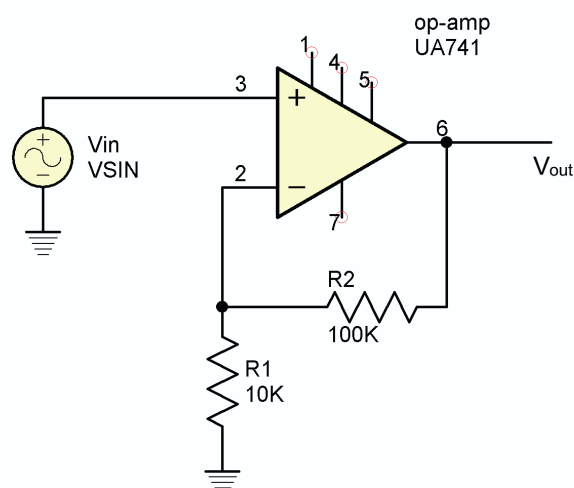



Figura 22 *Schema di principio di un amplificatore non-invertente con A.O.*

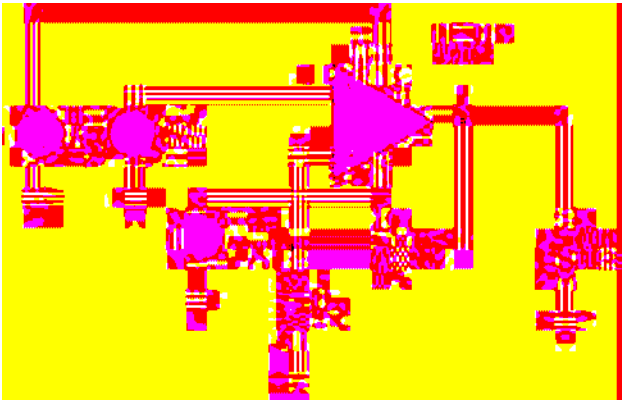



Figura 23 Circuito completo per amplificatore non-invertente con A.O.

Figura 24 Simulazione dell'amplificatore non-invertente di figura 23

Se però aumentiamo l'ampiezza di Vin fino a 2V le cose non vanno bene, infatti si avrebbe Vout(max) = 2V x 10 = 20 V.
L'amplificatore non può fornire in uscita una tensione di 20V in quanto è alimentato a 15V. La forma d'onda di uscita viene tosata (figura 21).
L'inconveniente di questo circuito è la sua resistenza di ingresso, che è pari ad R_1. Tale caratteristica indesiderabile viene superata con la prossima configurazione.

Amplificatore non-invertente

Il circuito di principio dell'amplificatore non-invertente è quello di figura 22. Applicando le due regole d'oro si ricava facilmente

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{in}$$

cioè il guadagno è

$$A_v = 1 + \frac{R_2}{R_1}$$

con le approssimazioni usate la resistenza di ingresso è infinita (è circa 10^{12} Ω se usiamo un LF411, oppure superiore a 10^8 Ω con un componente bipolare).
Come si vede dalla formula di Av, questa volta l'amplificatore non sfasa. Supponiamo R_1= 10 kΩ ed R_2=100 kΩ, quindi Av= 11; in figura 23 è mostrato il circuito completo per la simulazione, mentre in figura 24 è riportata la simulazione per un segnale di ingresso con V_M= 1 V ed f = 1 kHz; si noti che non esiste sfasamento tra segnale di ingresso e di uscita.

PROSSIMA PUNTATA

La prossima volta sperimenteremo alcuni semplici circuiti applicativi con amplificatori operazionali. Sarà fondamentale il simulatore Cadlogix, che ci aiuterà nella comprensione del loro funzionamento.

BIBLIOGRAFIA

Un capitolo meraviglioso sugli amplificatori operazionali si trova in:

1. P. Horowitz, W. Hill – "The Art of Electronics", Cambridge University Press, II ed., 1989

Una buona trattazione è anche contenuta in:

2. J. Millman, A. Grabel – „Microelectronics", McGraw-Hill International Ed., II Ed., 1987

Un riferimento irrinunciabile sono in questo caso i manuali dei costruttori; segnalo in particolare:

3. Texas Instruments – "Amplifiers and Comparators Data Book", 2000 che contiene numerose appendici riguardanti le definizioni dei parametri elettrici, nonché numerosi circuiti applicativi commentati.

Datasheet dei componenti sono disponibili presso le case costruttrici (nel sito della rivista ve ne sono alcuni):

4. Philips Semiconductor www.semiconductors.philips.com
5. ON Semiconductor - http://onsemi.com
6. Fairchild Semiconductor www.fairchildsemi.com
7. National Semiconductor - www.national.com
8. Analog Devices - www.analog.com
9. Texas Instruments - www.ti.com

Intelligenza artificiale

In questa puntata considereremo l'impiego della logica fuzzy per realizzare un semplice controllo di guida per un robot autonomo. Verranno analizzati i dettagli dell'implementazione e commentato il codice che implementa il controllore.

DESCRIZIONE DEL PROBLEMA

Ci proponiamo di realizzare un sistema di guida per un robot, che riesca a farlo muovere all'interno di un percorso delimitato da ostacoli o apposite segnalazioni. Un esempio di questa situazione potrebbe essere rappresentato da una piccola auto da corsa che deve percorrere autonomamente un circuito, o da un robot che deve muoversi all'interno dei corridoi di un edificio. In entrambi i casi il robot deve procedere in avanti seguendo il percorso, e mantenersi sufficientemente lontano dalle pareti o dai bordi. Per individuare i bordi del percorso il robot utilizza due sensori di distanza, che potrebbero essere ad esempio ad ultrasuoni o a raggi infrarossi a seconda delle sue dimensioni e della larghezza media del percorso. I sensori sono posti ai due lati del robot, puntati a poco meno di 90° rispetto la direzione i marcia (risultano quindi leggermente inclinati in avanti), come mostrato in figura 1. L'idea che sta alla base del controllo di guida è quella di cambiare la direzione di marcia in modo da mantenersi sempre equidistante dai due bordi del percorso.
Il problema non presenta particolari difficoltà in questa versione, ed è possibile ottenere il risultato voluto utilizzando tecniche di controllo classiche e semplici algoritmi, tuttavia in questo articolo proveremo a risolverlo utilizzando la logica fuzzy (rimando all'articolo pubblicato due numeri fa per maggiori dettagli sulla teoria). Come già spiegato in uno degli scorsi articoli il vantaggio principale della logica fuzzy non risiede tanto in un miglioramento delle prestazioni, ma in una più semplice descrizione del comportamento voluto, soprattutto quando questo non è facilmente modellizzabile matematicamente. L'utilizzo della logica fuzzy in questo caso risulta molto conveniente se ad esempio si vuole controllare in funzione dei dati in ingresso anche la velocità oltre che l'angolo di sterzata, in modo da massimizzare la velocità di percorrenza del circuito o minimizzare la probabilità di urto con le pareti. Le relazioni matematiche che intercorrono tra queste grandezze non sono facilmente intuibili, ma è abbastanza facile immaginare un'insieme di regole che descrivano bene questo comportamento.
Limitandoci per semplicità al caso iniziale: il controllore dovrà quindi essere dotato di due ingressi, relativi alla distanza rilevata dai due sensori, ed una uscita, proporzionale all'angolo di sterzata. Supponiamo che la velocità, dopo una prima fase di accelerazione, si mantenga approssimativamente costante.

DESCRIZIONE DEL CONTROLLORE

Come già detto possiamo pensare al nostro controllore come una *black box* dotata di due ingressi, ed un uscita. Supponiamo che i segnali di ingresso al controllore, che indicano la distanza dal bordo sinistro e destro, siano disponibili in forma digitale codificati come numeri ad 8 bit senza segno (quindi numeri da 0 a 255). L'unità di misura con cui sono espresse le distanze non è rilevante, è sufficiente che la portata dei sensori sia para-

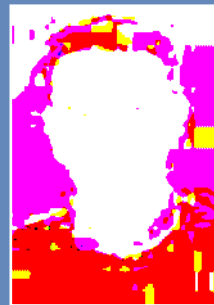
Controllo basato su Logica Fuzzy

di Antonio Di Stefano
a.distefano@farelettronica.com

Figura 1 Puntamento dei sensori di distanza (in blu)

gonabile alla larghezza del percorso (possibilmente maggiore). È anche necessario che se la distanza da misurare risulta superiore alla portata del sensore, questi restituisca comunque il valore massimo (e non un valore casuale o altre indicazioni di errore).
L'uscita del controllore esprime l'angolo di sterzata da utilizzare. Anche in questo caso si può utilizzare un valore ad 8 bit, questa volta con segno (quindi un numero da -128 a 127), che verrà utilizzato da apposite routine per comandare i motori in modo da ottenere un effetto proporzionale sulle ruote del robot. Dal momento che il numero è dotato di segno si possono considerare i valori positivi come una deviazione verso destra, mentre i valori negativi verso sinistra. Un valore prossimo a 0 indica invece che la direzione non deve essere cambiata.
Per realizzare il controllore fuzzy, una volta stabiliti gli ingressi e le uscite, e necessario definire l'insieme dei termini linguistici che vogliamo utilizzare per rappresentare le varie grandezze, e definirne le rispettive funzioni di appartenenza, e le regole da utilizzare.

Funzioni di appartenenza

Come già detto gli ingressi del controllore rappresentano delle distanze, è possibile quindi utilizzare 3 termini linguistici per descrivere le quantità corrispondenti: "Piccola", "Media", "Grande". Sarebbe possibile utilizzare un numero maggiore di termini, ottenendo così un comportamento più continuo e regolare, però occorrerebbe definire un numero maggiore di funzioni di appartenenza e di regole. L'uso di soli tre termini dà già risultati abbastanza soddisfacenti. Le funzioni di appartenenza (*Membership Functions*) relative ai tre termini sono state scelte di forma triangolare simmetrica, e sono rappresentate in figura 2. Ricordiamo che il grafico rappresenta in ascissa il dato numerico di ingresso, ed in ordinata il valore di appartenenza ad uno dei sottoinsiemi. Come si può vedere le tre funzioni sono state scelte in modo da sovrapporsi parzialmente, e da risultare simmetriche rispetto al centro.
Per rappresentare gli insiemi relativi all'uscita si è proceduto in maniera un po' diversa rispetto a quanto spiegato nello scorso articolo. Anziché utilizzare delle funzioni di appartenenza di forma triangolare o trapezoidale, si è scelto di ricorrere direttamente a dei valori numerici

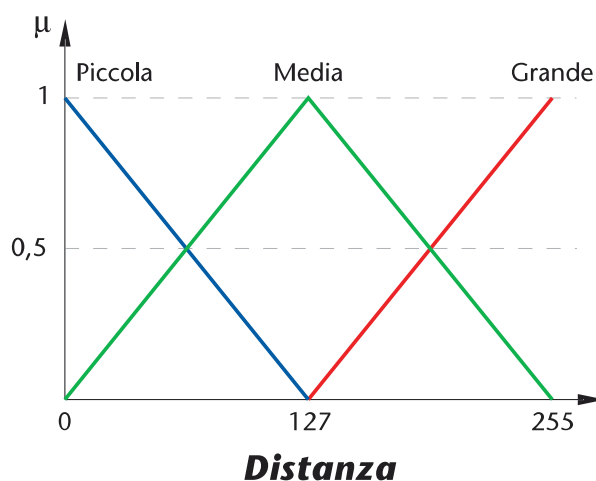



Figura 2 Funzioni di appartenenza degli ingressi (distanza)

(figura 3). Questa scelta se vogliamo è equivalente ad associare ad un certo numero di termini linguistici delle funzioni di appartenenza di forma molto sottile (al limite di larghezza 0), centrate proprio sui valori numerici voluti. Questo tipo di insieme è chiamato "*fuzzy singleton*", ed è comunemente utilizzato nell'implementazione di controllori fuzzy perché permette di eseguire in maniera più veloce i calcoli relativi alla fase di aggregazione e "defuzzificazione".

Regole

Le regole che sceglieremo devono fare in modo che il nostro robot segua il percorso e si mantenga adeguatamente distante dai bordi del tracciato. Intuitivamente questo si ottiene assicurandosi che la distanza rilevata dai due sensori sia il più simile possibile, ossia scegliendo di sterzare nella direzione della distanza più grande che abbiamo rilevato.

Seguendo questo ragionamento, le regole che utilizzeremo saranno del tipo "se la distanza di destra è *piccola* e la distanza di sinistra è *grande* allora l'angolo di sterzata è *grande verso sinistra*", oppure "se la distanza di destra è *media* e la distanza di sinistra è *piccola* allora l'angolo di sterzata è *piccolo verso destra*". Considerando tutte le combinazioni dei termini relativi agli ingressi è possibile scrivere le seguenti regole:

```
SE "dist. dx" è "Piccola" E "dist. sx" è
"Piccola" ALLORA "algolo" = 0
SE "dist. dx" è "Piccola" E "dist. sx" è
"Media"   ALLORA "algolo" = 10
SE "dist. dx" è "Piccola" E "dist. sx" è
"Grande"  ALLORA "algolo" = 100

SE "dist. dx" è "Media" E "dist. sx" è
"Piccola" ALLORA "algolo" = -10
SE "dist. dx" è "Media" E "dist. sx" è
"Media"   ALLORA "algolo" = 0
SE "dist. dx" è "Media" E "dist. sx" è
"Grande"  ALLORA "algolo" = 50

SE "dist. dx" è "Grande" E "dist. sx" è
"Piccola" ALLORA "algolo" = -100
SE "dist. dx" è "Grande" E "dist. sx" è
"Media"   ALLORA "algolo" = -50
SE "dist. dx" è "Grande" E "dist. sx" è
"Grande"  ALLORA "algolo" = 0
```

Queste regole possono essere espresse in forma sintetica utilizzando la "matrice delle regole", che è riportata nella tabella seguente.

dx\sx	Piccola	Media	Grande
Piccola	0	10	100
Media	-10	0	50
Grande	-100	-50	0

Il fatto che nella matrice delle regole troviamo dei valori numerici anziché dei termini linguistici è dovuto al fatto di avere scelto dei singleton per l'insieme di uscita, cioè gli esatti valori numerici che desidereremo avere in uscita per la regola corrispondente.

Ovviamente questo non significa che le uscite del controllore saranno limitate a questi valori, infatti quello che si verifica di solito è che più regole sono verificate contemporaneamente (ciascuna con il proprio grado di verità), e l'uscita risulterà una media pesata dei valori. Come abbiamo detto sono utilizzati valori ad 8 bit con segno, in cui i valori positivi corrispondono a un angolo di sterzata verso destra, mentre quelli negativi verso sinistra.

I valori utilizzati nella matrice delle regole sono stati scelti arbitrariamente, rispettando un criterio intuitivo di proporzionalità tra i gli effetti voluti per ciascuna regola.

Calcolo dell'uscita

Come spiegato nello scorso articolo per calcolare l'uscita numerica del controllore è necessario prima valutare tutte le regole, ricavando per cia-

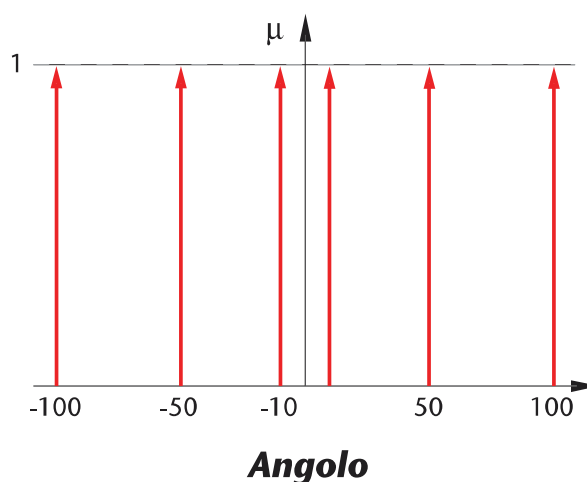



Figura 3 Funzioni di appartenenza dell'uscita (angolo di sterzata)

scuna l'uscita (c_i) e il suo grado di verità (w_i), e poi applicare il processo di aggregazione e defuzzificazione, che in questo caso si eseguono in un unico passo utilizzando la formula del "centro di massa":

$$\text{Uscita} = \Sigma_i (w_i * c_i) / \Sigma_i (w_i)$$

Se i calcoli sono stati effettuati correttamente anche l'uscita sarà un valore ad 8 bit con segno.

IMPLEMENTAZIONE DEL CONTROLLORE

Come nello scorso articolo progetteremo il controllore in modo da potere essere facilmente implementato su un microcontrollore ad 8 bit. Questo significa che tutte le grandezze e le variabili principali verranno rappresentate con numeri ad 8 o a 16 bit, e verrà utilizzata solamente l'aritmetica intera. Anche in questo caso il codice che verrà presentato sarà in Basic, per renderlo il più possibile intuitivo ed indipendente dall'hardware (verrà usata una sintassi simile a quella del Visual Basic, ma non perfettamente compatibile).

Fuzzificazione

Uno dei primi passi che un controllore fuzzy deve eseguire è quello di associare ad ogni grandezza in ingresso il valore di appartenenza ai vari termini linguistici utilizzati. Come già detto questo processo viene chiamato "fuzzificazione", e consiste sostanzialmente nel calcolare il valore assunto delle varie funzioni di appartenenza in corrispondenza del valore dell'ingresso corrente.
Nello scorso articolo è stato spiegato che un metodo molto rapido per eseguire questo calcolo è quello di creare delle tabelle in cui si riporta il valore delle funzioni di appartenenza per tutti (o un sottoinsieme) dei valori d'ingresso.
In questo caso, dal momento che la forma delle funzioni di appartenenza utilizzate lo consente, il valore verrà calcolato matematicamente utilizzando una semplice formula. Prima di vedere il codice che implementa la funzione è però necessaria un'ulteriore spiegazione. Nello scorso articolo abbiamo supposto che il valore assunto dalla funzione di appartenenza fosse un numero compreso tra 0 ed 1, quindi un numero reale, o comunque frazionario. Se, come nel nostro caso, si sceglie di utilizzare soltanto numeri interi è possibile esprimere i gradi di appartenenza utilizzando anche questi ultimi. Se decidiamo che il grado di appartenenza sia espresso da un numero ad 8 bit, allora esso avrà un valore compreso tra 0 e 255.
Dal punto di vista numerico questo non cambia il risultato che si ottiene in fase di defuzzificazione.
Il codice mostrato di seguito calcola il valore di appartenenza di una funzione di forma triangolare simmetrica di cui è possibile scegliere il centro e la larghezza della base:

```
Function MF(inp, cent, base) As Byte
  s = 255 - Abs(inp - cent) * base
  If s < 0 Then s = 0
  If s > 255 Then s = 255
  MF = s
End Function
```

Nella funzione il parametro *inp* è il valore numerico d'ingresso, *cent* la posizione del centro del triangolo, e *base* è un coefficiente correlato con la larghezza della base del triangolo. In pratica se *base* vale 1, il triangolo ha una base larga il doppio dell'intervallo d'ingresso, se vale 2 la base è larga quanto l'intero intervallo, se vale 4 la metà, e così via…
Tutti i parametri di ingresso, ed anche il valore restituito della funzione, sono dei numeri ad 8 bit senza segno.
Considerando le funzioni di appartenenza che vogliamo utilizzare per gli ingressi (figura 2), queste avranno centro 0, 127 e 255 rispettivamente, e base pari a 2.

Valutazione delle regole e defuzzificazione

Per valutare le regole, come spiegato nello scorso articolo, occorre prima di tutto calcolare il grado di appartenenza degli ingressi ai relativi termini linguistici, e poi calcolare l'AND fuzzy, che equivale a scegliere il minimo fra i due gradi di appartenenza.
Questo valore rappresenta il grado di verità dell'uscita associata alla regola. Dal momento che tutte le regole devono essere valutate, è necessario utilizzare un ciclo. Il codice seguente

implementa l'intero controllore fuzzy: esso riceve in ingresso i dati relativi alle distanze misurate, valutata tutte le regole (richiamando la funzione di fuzzificazione vista prima) e calcola l'uscita utilizzando la formula di defuzzificazione:

```
Function FuzzyDrive(ssx, sdx) As Byte

Dim wcacc As Integer
Dim wacc As Integer
Dim rules(2, 2) As Byte

' *** matrice delle regole ***
rules(0, 0) = 0
rules(0, 1) = 10
rules(0, 2) = 100

rules(1, 0) = -10
rules(1, 1) = 0
rules(1, 2) = 50

rules(2, 0) = -100
rules(2, 1) = -50
rules(2, 2) = 0

' *** Valutazione regole ***
wcacc = 0
wacc = 0
For s = 0 To 2
  For d = 0 To 2
    w = min(MF(ssx, s * 127, 2), MF(sdx, d
      * 127, 2))
    wcacc = wcacc + w * rules(s, d)
    wacc = wacc + w
  Next
Next

' *** Defuzzificazione ***
If wacc <> 0 Then
  FuzzyDrive = wcacc / wacc
Else
  FuzzyDrive = 0
End If

End Function
```

Tutti i valori d'ingresso ed uscita della funzione come già detto sono dei numeri ad 8 bit. Le principali variabili impiegate nei calcoli sono i due accumulatori utilizzati nella formula di defuzzificazione, che sono degli interi a 16 bit (con segno) e la matrice delle regole, dichiarata come una matrice bidimensionale (3x3) di valori ad 8 bit con segno.
Per valutare le regole viene utilizzato un doppio ciclo in cui vengono valutate tutte le combinazioni dei termini linguistici d'ingresso. Per ciascuna combinazione (quindi per ciascuna regola) viene calcolato il valore di appartenenza dei due ingressi al rispettivo termine linguistico (grazie alla funzione MF), scelto il minimo, ed accumulati i risultati parziali. Il risultato finale viene ottenuto grazie ad una divisione dei due prodotti parziali, secondo la formula del centro di massa vista prima. Prima di eseguire la divisione viene verificato che il denominatore non sia uguale a zero, il che risulterebbe in una condizione di errore (in questo caso si è scelto di fornire un'uscita "neutrale").
Una piccola nota riguardo alla moltiplicazione w*rules(s,d), che viene utilizzata all'interno del ciclo di valutazione delle regole. In questo caso non è stato usato nessun fattore di divisione per prevenire eventuali overflow.
Questo è reso possibile dai particolari valori utilizzati nella matrice delle regole, e dal fatto che al massimo 4 regole adiacenti possono essere verificate (cioè avere un w diverso da 0). In nessun caso quindi può verificarsi un overflow. Questa comunque è una condizione abbastanza particolare, che non è detto che sia verificata anche in altre applicazioni. In questi casi occorre dividere il prodotto per un opportuno coefficiente, come spiegato nelle scorse puntate.

CONSIDERAZIONI

Per utilizzare il controllore in una qualsiasi applicazione è sufficiente richiamare la funzione vista prima, passandole i due parametri relativi alle distanze misurate dai sensori. Il valore restituito dalla funzione può essere utilizzato per controllare i motori.
A tal proposito è necessario fare alcune precisazioni. Innanzi tutto per un corretto funzionamento del sistema è necessario che il valore degli ingressi forniti alla routine siano opportunamente "normalizzati". Bisogna cioè assi-

curarsi che l'intervallo di valori forniti dai sensori in un determinato percorso rientrino nel range che va da 0 a 255, e che si ottenga un valore prossimo a 127 quando il robot si trova equidistante dai bordi.
Questo è particolarmente importante perché le funzioni di appartenenza sono state costruite in modo che la distanza considerata "normale" abbai un valore centrale di 127. In altre parole, se il controllore riceve in ingresso valori molto più piccoli o molto più grandi in condizioni normali, non sarà in grado di interpretare correttamente i dati rilevati e quindi non si comporterà come previsto (una delle regole ad esempio dice che se entrambe le distanze sono "piccole" non si ha nessuna variazione di direzione; in condizioni normali questa condizione non dovrebbe essere verificata).
Un discorso analogo può essere fatto per il coefficiente di proporzionalità tra il valore fornito in uscita dal controllore e l'effettivo angolo di sterzata utilizzato. Se questo coefficiente viene scelto troppo grande si possono ottenere dei comportamenti instabili, se viene scelto troppo piccolo il robot può non riuscire a seguire correttamente le curve più strette. I termini "troppo piccolo" e "troppo grande" devono intendersi in relazione alla dimensione del tracciato ed alla velocità media del robot, e vanno quindi determinati di caso in caso.
In figura 4 è visibile il risultato di una simulazione, che è stata utilizzata per testare il corretto funzionamento del controllore su diversi tipi di percorsi, e con l'utilizzo di diversi parametri. Il tracciato rosso è la traiettoria seguita dal robot (partito dal punto 1).
Come si può vedere il controllore ha delle buone prestazioni anche in un percorso piuttosto articolato come quello mostrato in figura.
Il risultato visibile in figura è stato ottenuto utilizzando due sensori inclinati di circa 60° rispetto alla direzione di marcia, e che forniscono il valore massimo in uscita (255) in corrispondenza ad una distanza di 1.25 volte la larghezza media del tracciato. Nella parte bassa della figura sono visibili i segnali forniti dai sensori lungo il tracciato (Sx e Dx), e l'uscita del controllore in corrispondenza (ST).
Una volta realizzata questa versione del con-

Figura 4 Risultati della simulazione

trollore è possibile condurre qualche esperimento per comprendere come la scelta delle regole influenzi il comportamento e le prestazioni. Una modifica abbastanza interessante, e che fa apprezzare la flessibilità di un controllore fuzzy rispetto ad uno tradizionale, può essere la seguente: fare in modo che il robot svolti a sinistra o a destra agli incroci anziché proseguire dritto. Realizzare questo comportamento con un controllo classico non è proprio immediato, nel caso del controllore fuzzy invece basta modificare un dato nella matrice delle regole, in particolare nella regola corrispondente ad una distanza "grande" da entrambi i lati (condizione che si verifica proprio all'inizio degli incroci, o comunque in presenza di spazi "aperti").

Basta sostituire a 0 il valore 50 o 100 per ottenere il risultato voluto! In questo caso il robot seguirà le pareti alla sua destra quando non avrà più riferimenti davanti, anziché continuare dritto (figura 5). Un altro esperimento interessante consiste nell'utilizzare un altro controllore (quindi un'altra routine simile) per gestire la velocità del robot in funzione dei dati provenienti dagli stessi o da altri sensori. È semplice in questo caso ad esempio fare rallentare il robot in prossimità delle curve, e farlo accelerare nei rettilinei. Combinando i dati sarebbe perfino possibile creare una specie di "controllo di stabilità" (nel caso sia necessario percorrere le curve ad alta velocità…).

Figura 5 Simulazione eseguita dopo la modifica alla matrice delle regole

CONCLUSIONE

Si conclude qui questa serie di articoli dedicati all'Intelligenza Artificiale.

Il tentativo di trattare argomenti così vasti e complessi, e renderli facilmente comprensibili è stata una bella sfida! Spero di essere almeno riuscito nell'intento trasmettere una certa curiosità ed interesse per questi argomenti così affascinanti, ed ancora in parte inesplorati.

Per eventuali domande o dubbi sugli argomenti trattati è possibile contattarmi via e-mail.

Seconda parte
n° 240 - Giugno 2005
L'alternatore

Terza parte
n° 241/242 - Luglio/Agosto 2005
Costruzione delle pale

Quarta parte
n° 243 - Settembre 2005
Completamento e messa in opera

Costruire un generatore eolico partendo da zero

*Il sistema per raddrizzare la tensione di un alternatore pentafase è semplice, anche se poco usuale.
Si tratta di sistema realizzato utilizzando cinque ponti monofase da 25 Ampère equivalenti ad un totale di 20 diodi.*

IL RADDRIZZATORE

Lo schema elettrico è mostrato in figura 19 ed un possibile schema di montaggio in figura 20.

Ognuno dei ponti ha due ingressi per l'alternata e due uscite per la continua. Il cablaggio è semplice: basta collegare senza ordine prestabilito le dieci uscite dell'alternatore ai dieci ingressi di alternata dei ponti.
L'operazione è facilitata dall'uso di connettori FASTON. Per l'uscita in continua, abbiamo collegato tutti i positivi e tutti i negativi dei ponti rispettivamente a due morsetti per grandi correnti.
Da questi morsetti abbiamo fatto partire i due cavi della corrente continua che vanno agli accumulatori. Abbiamo montato i 5 ponti all'interno di un contenitore alettato in alluminio che si è dimostrato molto adatto allo scopo. Questo contenitore è stato montato posteriormente alla staffa dell'alternatore.

I due cavi di uscita, della sezione di 6 mm^2 (sezione minima 2,5 mm^2) della lunghezza necessaria per arrivare agli accumulatori, sono infilati all'interno del palo attraverso il foro praticato nel tappo di chiusura superiore del supporto girevole di imbardata.
È consigliabile limare i bordi taglienti del foro e proteggere l'entrata dei cavi con una guaina protettiva.

PERCHÉ UN ALTERNATORE PENTAFASE?

Nessuno vieta di costruire un generatore eolico monofase, che tra l'altro risulterebbe costruttivamente più semplice.
L'inconveniente di un generatore monofase è la rumorosa vibrazione trasmessa al palo di sostegno, superiore al fruscìo del vento sulle pale. Aumentando il numero delle fasi, la vibrazione

Figura 19 *Schema elettrico del raddrizzatore a cinque fasi*

Completamento e messa in opera

di Antonio Cecere
a.cecere@farelettronica.it

dell'alternatore diminuisce. La quasi totalità dei piccoli generatori eolici sono trifasi.

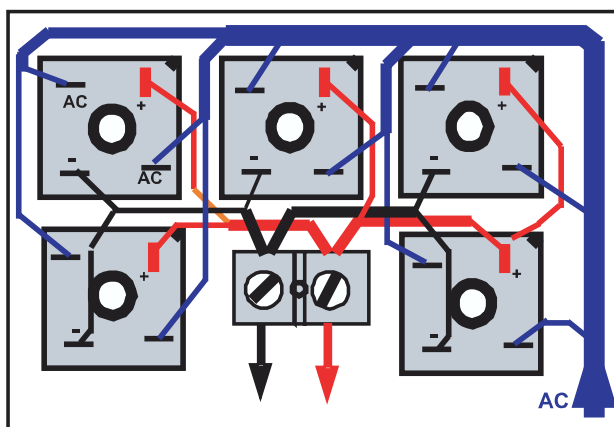



Figura 20 Disposizione dei cinque ponti in su un dissipatore di calore

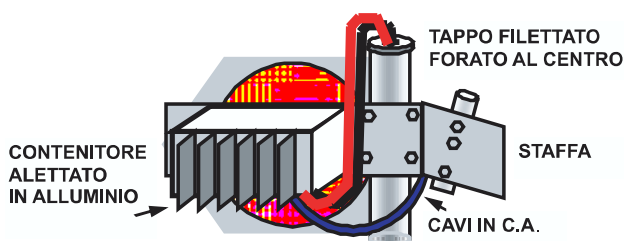



Figura 21 Montaggio del dissipatore

Foto 22 Dissipatore con i ponti raddrizzatori all'interno, montato dietro la staffa dell'alternatore

Un alternatore pentafase (una rarità) produce una vibrazione quasi impercettibile.

GLI ACCUMULATORI

La prima tentazione è quella di utilizzare gli accumulatori più facili da reperire, quelli per automobile. Ma, come spesso capita, la scelta più facile non è la migliore: infatti gli accumulatori per auto, essendo ottimizzati solo per correnti impulsive di breve durata (come le partenze) non sono adatti al nostro uso.
Esistono, anche se sono meno reperibili e più costosi, accumulatori appositamente costruiti per accumulo di energia (www.fiamm-gs.com) con vita media di 10 anni, contro i 3 anni degli accumulatori per auto.
Una delle caratteristiche poco conosciute degli accumulatori è la "**profondità di scarica**".
Perché gli accumulatori abbiano la vita dichiarata, essi non devono essere scaricati oltre una certa percentuale della capacità nominale: per gli accumulatori per auto la profondità di scarica ammessa è meno del 50%, invece per gli accumulatori per impianti di energia la scarica può arrivare all'80%. Poiché il vento non è sempre presente, è prudente che gli accumulatori abbiano un'autonomia da 3 a 5 giorni.
Per il generatore che stiamo costruendo, dovremmo avere un minimo di quattro accumulatori a 12 Volt da 100 Ah ciascuno, posti a due a due in serie e poi in parallelo per avere un sistema a 24V. In mancanza di batterie (la scuola è a corto di fondi e non ha sponsor), abbiamo provato il generatore simulando le batterie con un regolatore di carica di tipo shunt (vedi oltre).
Questo apparecchio assorbe corrente solo se viene superata una tensione stabilita, proprio come una batteria, con la differenza che consuma e non restituisce l'energia.

IL REGOLATORE DI CARICA

Il regolatore di carica fa in modo da evitare la

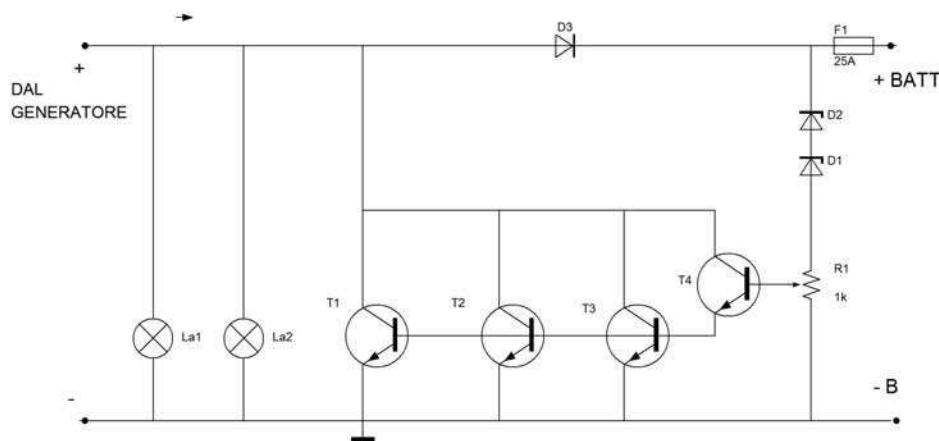



***Figura 22** Schema del carico con cui è stato provato il generatore, costituito da due lampade da 24V – 60W e da un regolatore shunt equivalente ad un grosso diodo zener di oltre 300W.*

sovraccarica degli accumulatori. La sovraccarica porta alla gassificazione dell'elettrolita e ad un rapido invecchiamento degli accumulatori.
Il regolatore deve essere di tipo shunt, cioè si deve porre in parallelo al generatore e deviare la corrente in eccesso su un carico "zavorra" che può essere una resistenza o un carico qualsiasi che abbia possibilmente qualche utilità.
Si deve evitare di lasciare il generatore scollegato: in questa condizione, non essendoci alcuna azione frenante da parte dello statore, i rotori e la pale assumerebbero una velocità pericolosa ed il generatore verrebbe sottoposto a fortissime sollecitazioni. È possibile invece lasciare il generatore in cortocircuito senza che esso si danneggi. Il cortocircuito blocca quasi del tutto la rotazione delle pale.

Il regolatore shunt è progettato in modo che, superata la tensione massima degli accumulatori (28,6 Volt per accumulatori a 24 Volt), un relè colleghi un carico di grossa potenza in parallelo al generatore.

PROTEZIONE DALLE VELOCITÀ ECCESSIVE

Esistono molti sistemi per proteggere un generatore eolico dai venti di forte intensità (oltre i 15 m/s). Noi abbiamo voluto riprodurre quello più utilizzato dagli hobbisti. Si tratta di un congegno passivo di tipo meccanico detto della "**coda girevole**". La coda di orientamento viene fissata alla staffa dell'alternatore per mezzo di uno snodo leggermente obliquo (20°) rispetto alla verticale.

Produttori di generatori eolici

Esistono due tipi sostanzialmente diversi di produttori: quelli di piccoli generatori (da 60W a 20 KW) e quelli di grandi generatori (da 20KW a 4 MW). I piccoli generatori sono adatti all'accumulo o allo scambio in rete per piccole e medie utenze. I grandi generatori servono esclusivamente alla produzione ed all'immissione in rete dell'energia prodotta.
Ecco alcuni produttori:

Piccoli generatori:
Gmdel, Salmini, Ropatec (Italia)
Vergnet, Eoltec (Francia)
Inclin, Solener, Bornay (Spagna)
Iskra, Marlec, Proven (Inghilterra)
Southwest, Bergey, WTIC Jacobs (USA)
African Wind Power (Zimbawe)

Grandi generatori:
Vestas, Neg-Micon, Nordex, Bonus (Danimarca)
Gamesa, Made (Spagna)
Enercon, Fuhrlander (Germania)
GE Wind Energy (USA)
Lagerwey (Olanda); Mitsubishi Electric (Giappone)
Windflow (Nuova Zelanda)

Figura 23 *Snodo della coda*

Con venti normali, la coda fa da contrappeso all'alternatore ed orienta le pale in modo che esse siano investite frontalmente dal vento. Con venti eccessivi la coda non è più sufficiente a mantenere la posizione delle pale. Queste, essendo disassate rispetto al supporto di imbardata, si pongono di traverso in modo tale da essere investite lateralmente dal vento e quindi da essere meno sollecitate.
La coda piegandosi acquista energia potenziale perché, a causa dell'angolazione dello snodo, il suo baricentro viene spostato più in alto. Quando il vento cala, la coda, perdendo l'energia potenziale, si riporta nella sua posizione iniziale e nello stesso tempo riporta anche le pale nella posizione perpendicolare al vento.
Il meccanismo non è di facile comprensione, se non aiutandosi con disegni e foto. L'esperienza di molti hobbisti conferma che, non avendo organi attivi, esso risulta molto affidabile.
La coda è stata realizzata con un tubo da 1", lungo 120 cm, che termina con una banderuola in bachelite (ma può essere usato un compensato idrorepellente) la cui forma è del tutto libera, ferma restando la superficie di circa 0,45 mq. Noi abbiamo realizzato una banderuola a forma di lampadina, per simboleggiare i contenuti del nostro corso (l'idea è del compagno Alessandro Laguardia).
Il tubo della coda opposto alla banderuola, terminante a pipa, viene infilato all'interno del tubo di snodo (Ø = 1 $^{1}/_{4}$") fissato obliquamente a 20 gradi dalla verticale sulla staffa (la stessa dell'alternatore).
La coda deve poter oscillare entro un angolo di circa 100 gradi.

IL CAVO

Il cavo di collegamento tra raddrizzatore ed accumulatori è fonte di perdita di potenza per effetto Joule ($Pp=R*I^2$). A 24 V, con una potenza di carica di 500W, usando una coppia di cavetti da 6 mm^2, la perdita di potenza su una linea di 60 metri è del 24%. La perdita scende al 16% se la sezione usata è di 10 mm^2.
Poiché il generatore si gira per orientarsi con il vento, non potrebbe il cavo di discesa

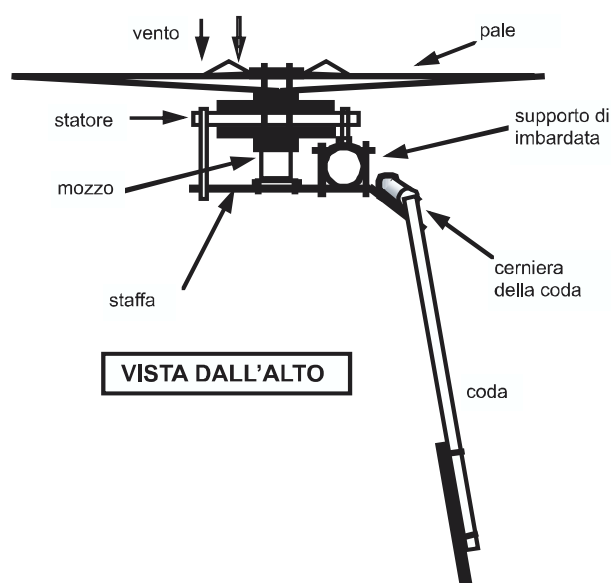



Figura 24 *Schema di montaggio del generatore con applicazione della coda girevole.*

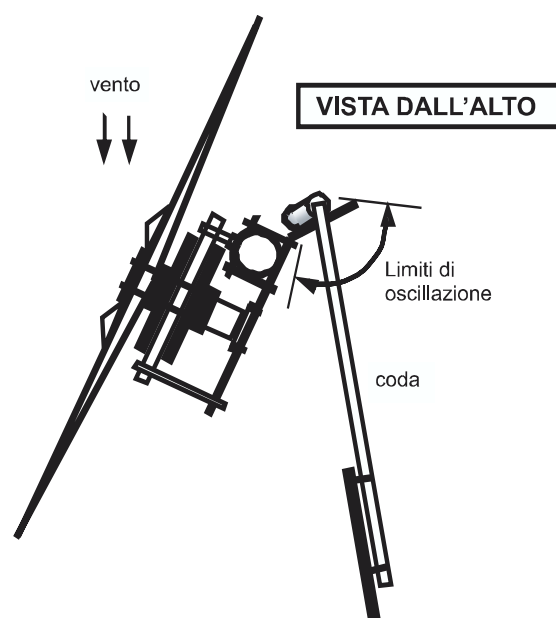



Figura 25 *Posizione della coda con venti eccessivi*

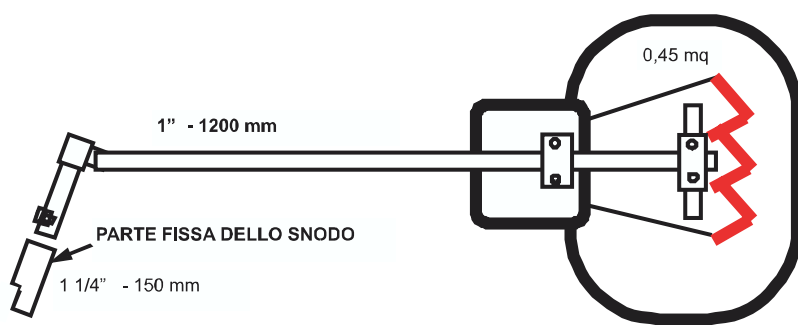



Figura 26 Coda del generatore eolico con attacco snodato e banderuola

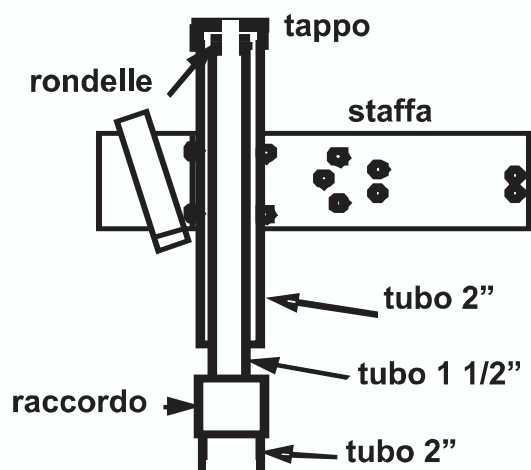



Figura 27 Montaggio dell'alternatore a testa-palo

Foto 23 Raccordo filettato per adattare tubo da 1,5" a quello da 2"

attorcigliarsi e rompersi?
Pur essendo teoricamente possibile, non è mai successo che il cavo pendente all'interno del palo di sostegno si sia rotto per essersi attorcigliato. Pochi si sono imbarcati nella sperimentazione di un meccanismo a spazzole, che nella pratica ha creato più problemi di quelli che doveva risolvere.

IL MONTAGGIO DEL PALO

In genere si scelgono come pali dei tubi di acciaio per idraulica di 2 o 3 pollici. La sommità del palo deve avere un diametro di 1,5", per poterci sovrapporre a "testa-palo" il supporto girevole di imbardata (che regge coda e alternatore).

Lo spezzone finale da 1,5" viene di solito saldato sul palo da 2", ma noi abbiamo preferito utilizzare un raccordo filettato.
Issare un palo è semplice se si dispone di una gru, un po' più difficile per degli appassionati che si trovano in posti poco accessibili. In generale si sceglie di tenere il palo in verticale con dei **tiranti in acciaio da 4 mm** di diametro.
Il palo può essere tirato su con l'aiuto di due o tre persone e applicando ad esso in via provvisoria un paletto in perpendicolare nella parte bassa, per allungare il "braccio" della forza con cui si innalza il palo rispetto al punto di appoggio a terra (vedi figura 28).
In genere il palo non viene incastrato nel punto di appoggio, ma solo incernierato, facilitando così l'eventuale abbassamento per la manutenzione dell'alternatore e delle pale. I tiranti devono essere almeno quattro e, se il palo supera i 10 metri, è consigliabile applicarne 8 o 12 a varie altezze. L'operazione non è priva di rischi: occorre stare molto attenti a non far cadere il palo mentre lo si alza. I due cavi di uscita devono essere cortocircuitati per evitare che il generatore vada in folle quando lo si sta alzando.
Nel nostro caso noi abbiamo montato il generatore nell'unico sito per noi disponibile… il terrazzo della scuola. Non è il sito migliore, ma non abbiamo altre scelte.

Abbiamo montato il palo da 2" come prolunga di un traliccio per pallavolo, ottenendo una lunghezza totale di 7 metri. L'altezza non è ottimale, ma potremo verificare il funzionamento nel tempo. Finora abbiamo fatto delle prove di funzionamento su banco per far accendere delle lampade ed ora aspettiamo delle giornate di vento buono per fare le verifiche sul campo.

COMMENTO FINALE

Questo è il nostro ultimo anno di corso e per molti di noi c'è già un lavoro che ci aspetta. Finita la scuola, sarà difficile incontrarci di nuovo tutti insieme e continuare a seguire il nostro generatore.
I ricordi della scuola diventeranno sempre più

sbiaditi, ma non dimenticheremo questa esperienza esaltante con cui insieme siamo entrati nella storia… o no? Per quello che sappiamo, questa è la prima esperienza di questo genere in Italia.
Sappiamo di non essere stati perfetti e che il nostro lavoro è migliorabile, ma siamo felici di aver aperto una strada e di aver dato molte indicazioni a chi vorrà continuare da solo.
Ringraziamo i nostri compagni del settore meccanico che hanno collaborato quando ci mancavano attrezzi e competenze.
Ringraziamo il sig. Hugh Piggott, che non ci ha mai fatto mancare il supporto della sua esperienza via e-mail dalla Scozia.
Ringraziamo infine il nostro docente di elettrotecnica, prof. Antonio Cecere, che ci ha guidato nei meandri della tecnica ed ha messo in ordine questo scritto, e la nostra docente di Inglese, prof. Beatrice Paparella, che ci ha aiutato a tradurre tutti i documenti in inglese presi da INTERNET: senza di loro non saremmo riusciti a portare a termine questo progetto e siamo sicuri che essi continueranno a seguire la nostra creatura.
Ad essi abbiamo chiesto di restare a vostra disposizione e rispondere a tutti i vostri quesiti all'indirizzo e-mail: a.cecere@farelettronica.com.
Gli studenti che hanno partecipato a questo lavoro, ai quali vanno i nostri complimenti, sono: Pierangelo, Vito, Gaetano, Piero, Carlo, Giorgio, Donato, Valerio, Costantino, Mario e Gianpaolo.

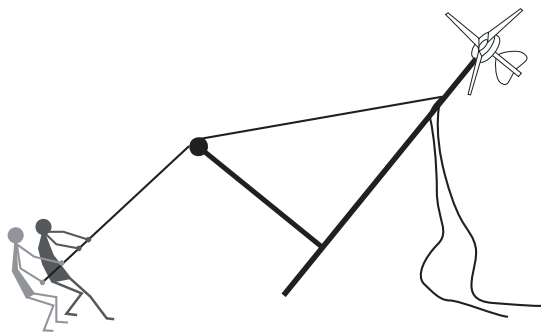



Figura 28 *Sistema di sollevamento di un palo per generatore eolico*

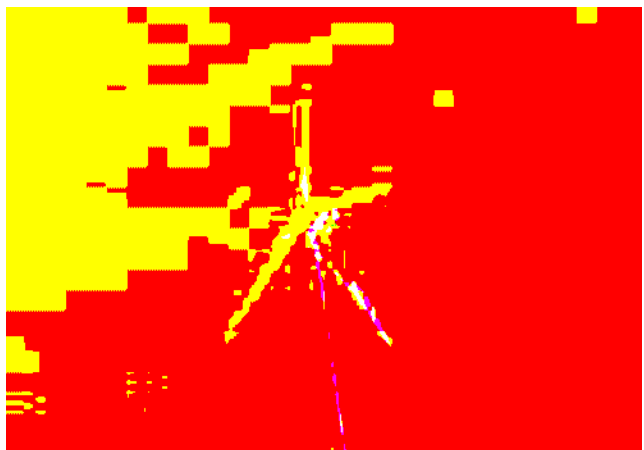
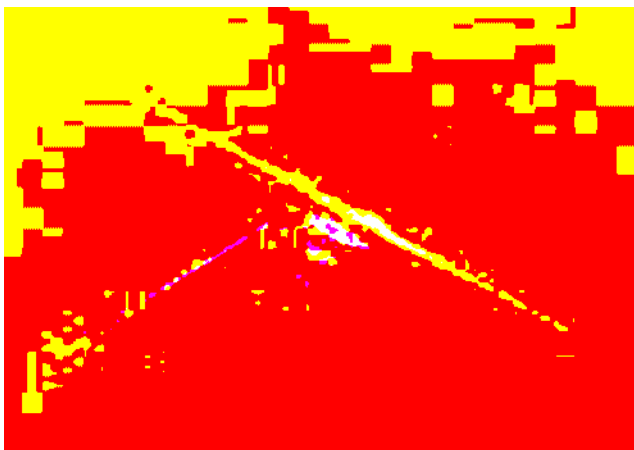
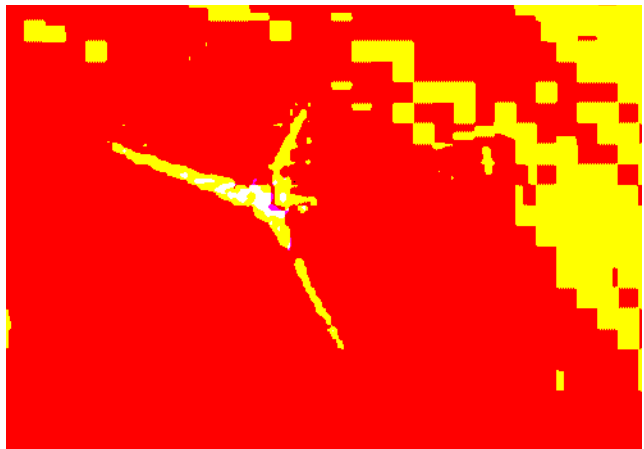



Foto 23 *Il montaggio del generatore è terminato, ora aspettiamo il vento*

MODULO DI GESTIONE RF PER CONTROLLO ACCESSI RFID

SCHEDA DI CONTROLLO PER CONTROLLO ACCESSI RFID

ANTENNA 125 KHZ PER CONTROLLO ACCESSI

TRANSPONDER ATTIVO RFID

Come nasce un circuito

Oramai i circuiti integrati sono parte di tutti i progetti, anche i più semplici. Ma cosa c'è dietro un circuito integrato? In che modo è possibile inserire migliaia e migliaia di transistors su una "scheggia" di silicio di appena qualche millimetro quadrato? Cosa contiene quella piccola scatoletta nera che assomiglia più ad un insetto che ad un gioiello di tecnologia? Iniziamo il nostro viaggio per scoprire i misteri di questo mondo veramente affascinante

La materia prima è il silicio, un elemento della tavola periodica noto per le sue caratteristiche di semiconduttore. Le sue proprietà conduttive variano infatti con la temperatura, ma se viene opportunamente "inquinato" con altri elementi diviene l'elemento base di tutta l'elettronica integrata. Con la grande evoluzione della tecnologia, è possibile costruire su una scheggia (chip) di pochi millimetri quadrati, un gran numero di dispositivi come transistor, diodi, resistenze, condensatori e persino microstrutture dall'aspetto veramente suggestivo. Ma procediamo con ordine e partiamo dall'inizio…

UN GRANELLO DI SABBIA

Per essere utilizzato in elettronica, il silicio deve trovarsi in forma cristallina ed avere un elevato grado di purezza (1 parte su un miliardo). Il problema è che in natura non esiste silicio con queste caratteristiche, ma abbonda invece l'ossido di silicio (SiO_2 comunemente noto anche come silice o quarzo) che si trova in grandi quantità nelle spiagge marine. Sono dunque necessari diversi processi chimici a cui sottoporre i granelli di sabbia per ottenere un monocristallo purissimo adatto allo scopo.

Le prime reazioni chimiche

Il primo trattamento a cui viene sottoposto l'ossido di silicio, è una reazione chimica con acido cloridrico (HCl) che permette di trasformare l'ossido in una sostanza detta "triclorosilano" ($SiHCl_3$). La reazione avviene a 300°C ed il risultato è idrogeno gassoso ed appunto triclorosilano, che è in forma liquida.
Mediante distillazione frazionata, la sostanza ottenuta viene purificata e sottoposta alla reazione chimica inversa (quindi con idrogeno gassoso) per ottenere nuovamente silicio e acido cloridrico. Il silicio ottenuto in questa fase ha un grado di purezza idoneo all'impiego in microelettronica, ma ha il difetto di essere amorfo, ovvero di non avere una struttura cristallina.
Per ottenere un monocristallo esistono due metodi: il metodo Czochralsky ed il metodo FZ (Floating Zone).

Il metodo Czochralsky (CZ)

Il risultato di questa operazione è una barra cilin-

Figura 1 *Struttura per l'applicazione del metodo Czochralsky*

integrato

di Maurizio Del Corso
m.delcorso@farelettronica.com

drica del diametro di venti/trenta centimetri, costituita da silicio puro in forma monocristallina.
Questa tecnica prevede l'inserimento del silicio puro amorfo in un crogiuolo di quarzo rivestito esternamente di carbonio. Il tutto viene portato ad una temperatura di 1420°C per fondere il silicio amorfo in un ambiente riempito di gas inerte (Argon) in modo da prevenire ogni possibile contaminazione. In figura 1 è riportato lo schema della struttura.

La formazione del monocristallo avviene immergendo nel fuso un seme cristallino (un pezzo di silicio cristallino). Il seme viene lentamente tirato fuori dal fuso in modo che il silicio possa solidificarsi intorno ad esso e, durante la solidificazione, gli atomi si disporranno in maniera regolare seguendo la struttura cristallina del seme. Per favorire questo fenomeno, il seme viene fatto ruotare lentamente e la combinazione tra la velocità di rotazione e quella di tiraggio, determina il diametro della barra cristallina cresciuta.

Figura 2 *Fasi della crescita del monocristallo*

Il metodo FZ

Il metodo FZ (Floating Zone) permette di ottenere una barra cilindrica di silicio monocristallino con un grado di purezza maggiore rispetto al metodo Czochralsky ed un diametro minore (una decina di centimetri). La figura 3 mostra il principio di funzionamento. Un riscaldatore a radio frequenza fonde una specifica sezione del cilindro di silicio amorfo, che viene fatto ruotare e traslare lentamente in modo che la zona fusa, raffreddandosi consenta agli atomi di disporsi secondo la struttura cristallina imposta dal seme. Tutta l'operazione avviene in condizioni di vuoto spinto o in un ambiente riempito di gas inerte. Il cristallo ottenuto con questo metodo è di qualità migliore rispetto a quello ottenuto col metodo CZ in quanto non c'è l'interazione con il crogiuolo che è comunque fonte di impurità. Il metodo è detto Floating Zone proprio perché viene fusa via via solamente una zona della barra di silicio amorfo. In figura 4 è mostrato il risultato di questa tecnica.

Wafer

La barra di silicio cristallino ottenuta con uno dei due metodi visti in precedenza, viene lavorata in modo che il suo diametro sia costante per tutta la lunghezza, quindi ne viene analizzata l'orientazione cristallografica e identificata sulla barra stessa mediante la fresatura di due piani longitudinali seguendo un opportuno codice. Il lingotto così ottenuto viene tagliato in fette dello spessore di circa un millimetro (wafer) che vengono accura-

Figura 3 *Il metodo FZ*

tamente lucidate mediante l'operazione di lappatura con polvere di diamante e, successivamente, lucidatura in soda caustica: il risultato è un wafer lucidato perfettamente a specchio, la cui superficie ha una rugosità inferiore al milionesimo di metro.

Figura 4 *Una barra di silicio monocristallino ottenuta col metodo FZ*

LA REALIZZAZIONE DEI DISPOSITIVI

I dispositivi elettronici (diodi, transistor, resistenze, ecc…) sono costituiti, in generale, dalla giunzione di una o più zone di semiconduttore a diverso drogaggio. Il drogaggio avviene impiantando atomi di impurezze all'interno del cristallo e, a seconda delle caratteristiche della sostanza drogante utilizzata, si possono ottenere zone a drogaggio di tipo *p* o di tipo *n*.
Ad esempio un diodo è costituito dall'unione di una zona n ed una zona p, mentre un transistor bipolare è costituito dall'unione di tre zone n-p-n (o p-n-p). L'operazione di drogaggio avviene "sparando" gli atomi di impurezza direttamente sulle zone interessate, per cui è necessario utilizzare delle maschere in modo da proteggere le zone non interessate.
Il materiale utilizzato per la mascheratura delle zone è l'ossido di silicio che può essere ottenuto facilmente sottoponendo il wafer a condizioni particolari. Inoltre l'ossido è facilmente rimovibile utilizzando acido cloridrico al quale invece resiste perfettamente il silicio.

Ossidazione termica

L'ossidazione termica è il processo grazie al quale è possibile ricoprire il wafer con uno strato di ossido. Concettualmente l'ossidazione del wafer è molto simile al processo mediante il quale un comune pezzo di ferro, esposto in un ambiente umido, si ricopre di ruggine (che appunto è ossido di ferro).
Il processo di ossidazione termica permette di ottenere un ossido di ottima qualità adatto anche ad isolare le diverse piste conduttive all'interno del circuito integrato. L'ossidazione avviene all'interno di una fornace ad alta temperatura in ambiente ricco di ossigeno e vapore acqueo in cui vengono immerse le fette di silicio (figura 5). Le temperature in gioco variano tra i 900°C ed i 1200°C e tutto il processo è controllato da un complesso sistema a calcolatore.

Litografia

Al termine del processo di ossidazione, il wafer è interamente ricoperto di uno strato di ossido di silicio. Poiché alcune zone andranno drogate con impurezze di tipo *p* ed altre con impurezze di tipo *n*, è necessario rimuovere l'ossido in corrispondenza delle zone da drogare. La rimozione dell'ossido avviene mediante un attacco chimico seguendo un processo litografico del tutto simile a quello usato per la realizzazione dei circuiti stampati. Sul wafer viene infatti deposto uno strato di fotoresist (una sostanza sensibile ai raggi UV), ponendo il wafer in rapida rotazione (4000 giri/min circa) e colando un minimo quantitativo di sostanza fotosensibile.
I wafer così preparati vengono posti in un forno per una cottura leggera del resist (soft-bake) al fine di far evaporare tutto il solvente in cui è disciolto l'elemento fotosensibile.
Il processo successivo è quello dell'esposizione dei wafer alla radiazione ultravioletta. L'esposizione avviene attraverso delle maschere in cromo (equivalenti al master dei circuiti stampati).
Per i resist positivi, i raggi UV sono in grado di indebolire i legami tra i polimeri che compongono il materiale. Per quelli negativi si verifica il contrario: la radiazione UV rafforza i legami. Nel primo caso sulle maschere è necessario un disegno al positivo mentre nel secondo il disegno deve essere il negativo di ciò che si vuole riprodurre.
I wafer con resist esposti vengono immersi in un bagno a base di NaOH (idrossido di sodio) in grado di rimuovere il polimero più debole. Successivamente si effettua una cottura a 120 °C (hard-bake) per rendere il film resistente alle successive fasi di lavorazione.
Per rimuovere l'ossido non protetto dal resist, è

Figura 5 *Ossidazione dei wafer*

necessario un attacco chimico (etching) in acido fluoridrico. L’etching può avvenire per immersione (wet) o in fase vapore (dry). Nel primo caso l’attacco procederà in modo anisosotropo riuscendo ad eliminare anche parte dell’ossido sottostante al resist in corrispondenza dei bordi delle geometrie definite. Con questo tipo di processo vengono alterate le dimensioni delle zone aperte anche se di ciò si può facilmente tenere conto dimensionando opportunamente i disegni sulle maschere. L’attacco in fase vapore permette di ovviare a questo inconveniente. I wafer sono posti in una camera in cui, tramite l’applicazione di un campo elettrico in continua o in alternata, viene dissociato il gas reattivo utile alla rimozione dell’ossido. Anche nel caso di dry-etch vi sono comunque problemi legati alla non idealità del sistema: un attacco chimico più o meno prolungato nel tempo dà luogo ad aperture nell’ossido di dimensioni diverse dal disegno voluto. Il risultato del processo di litografia è un wafer ricoperto di ossido in cui sono aperte delle “finestre” (figura 6).

Drogaggio

Con la mascheratura di ossido di silicio, solo attraverso le zone scoperte può avvenire il drogaggio del silicio. I metodi impiegati sono: diffusione e impiantazione ionica.

Nel drogaggio per diffusione i wafer sono posti in una fornace (simile a quella utilizzata per l’ossidazione) in cui vengono però immessi gas contenenti l’elemento drogante. Le temperature di processo variano tra gli 800 e i 1200 °C.

Il drogante (che può essere un composto a base di Boro, Arsenico, Fosforo) penetra nelle finestre dell’ossido raggiungendo la superficie del wafer un po’ come una goccia di inchiostro si espande su un foglio di carta assorbente. È evidente quindi che il processo di diffusione può avvenire nel solido lungo qualunque direzione. In particolare, vi

Figura 6 *Un possibile risultato del processo litografico*

sarà diffusione di drogante nelle zone di confine della mascheratura di ossido.

La tecnica di drogaggio largamente utilizzata nella moderna tecnologia VLSI è quella per impiantazione. Con l’applicazione di un intenso campo elettrico, vengono creati ioni di sostanze droganti che, opportunamente selezionati con una fenditura, vengono diretti verso il bersaglio in una camera di accelerazione finale e raggiungono la superficie del wafer con una energia dell’ordine di qualche centinaio di KV che determina la profondità del drogaggio. È evidente che questa tecnica di

Figura 7 *Realizzazione di una zona p*

impiantazione per sua natura porta ad un disordine reticolare del substrato. Il danneggiamento superficiale degrada notevolmente le proprietà di trasporto elettronico nel wafer, quindi si rende necessario procedere ad un'opera di "rinvenimento" (annealing) ottenuta scaldando il substrato a temperature piuttosto elevate. Al termine del drogaggio, le zone del wafer non protette dall'ossido divengono zone *p* o *n* a seconda del drogante impiegato. A questo punto viene rimosso l'ossido, quindi si ripete il processo di ossidazione, litografia e drogaggio per ottenere sul wafer altre zone drogate.

In figura 8 è riportato un wafer su cui sono stati realizzati i dispositivi elettronici mediante i processi appena descritti.

IL DIODO

Un diodo è una giunzione p-n per cui la sua realizzazione su un substrato di silicio n comporta la realizzazione di una zona p e dei relativi contatti elettrici. La figura 9 mostra le fasi di realizzazione del diodo (ovviamente l'apertura di ciascuna finestra nell'ossido, comporta un intero processo litografico).

IL TRANSISTOR BJT

Per la realizzazione di un transistor bipolare a giunzione (ad esempio un pnp) partendo da un substrato di tipo n, è necessario ricavare le due giunzioni con tre operazioni di drogaggio. Il processo per la realizzazione di un pnp è schematizzato nella figura 10.

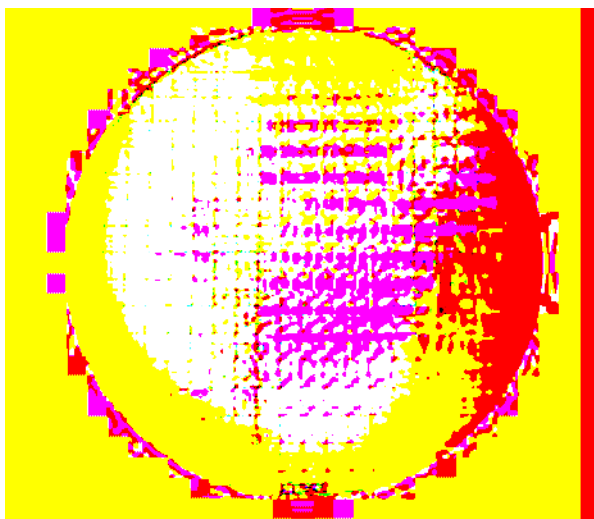



Figura 8 Un wafer al termine della realizzazione dei dispositivi

PULIZIA ESTREMA!

Tutte le operazioni di manipolazione del wafer ed in particolar modo la fase di esposizione del resist ai raggi UV, vengono condotte in ambienti estremamente puliti noti come "camere bianche". Le camere bianche di lavorazione hanno un livello di pulizia 1000 volte migliore di quello che si ha in una sala operatoria ospedaliera e prima dell'ingresso nella camera bianca, l'operatore deve indossare un vestiario che protegga completamente i wafer da qualunque contaminazione e sottoporsi ad una doccia d'aria che elimini ogni presenza di pulviscolo dal camice indossato.
Perché questa ossessione maniacale per la pulizia? La risposta è semplice: i dispositivi realizzati sul wafer hanno dimensioni inferiori al µm (1µm=0,000001m=0,001mm) per cui un granello di polvere di due millesimi di millimetro è sufficiente a danneggiare un intero dispositivo. La figura 11 mostra la fotografia al microscopio elettronico di un granello di impurezza su una pista di 1µm.

Figura 9 Realizzazione di un diodo a giunzione

INCAPSULAMENTO

Una volta ultimati tutti i processi di produzione dei dispositivi, il wafer appare come in figura 8. Segue una fase di test in cui viene accertato il corretto funzionamento dei chip realizzati e l'eventuale marcatura di quelli non funzionanti o non rispondenti alle specifiche.
Successivamente, mediante un processo di taglio, tutti i chip realizzati sul wafer vengono tagliati, quindi incapsulati in un contenitore di resina cui fanno capo i terminali per la gestione dei segnali (figura 12). Questo è l'aspetto tipico del circuito integrato che tutti conoscono.

Una fase molto delicata del processo di packaging è il bonding, ovvero la saldatura dei pad del chip ai pin del contenitore. La saldatura è fatta mediante una microscopica punta ad ultrasuoni. Nella figura 13 si può osservare il chip incollato sul proprio supporto e i fili conduttori sottilissimi che collegano i pad del chip ai pin del package.

TIPI DI PACKAGES

Per concludere la trattazione sui circuiti integrati, è doveroso accennare ai diversi tipi di packages utilizzati per l'incapsulamento del chip vero e proprio. I vari tipi di contenitori si differenziano per dimensioni, numero di pin, capacità di smaltimento del calore e tipo di montaggio. Si parla di dispo-

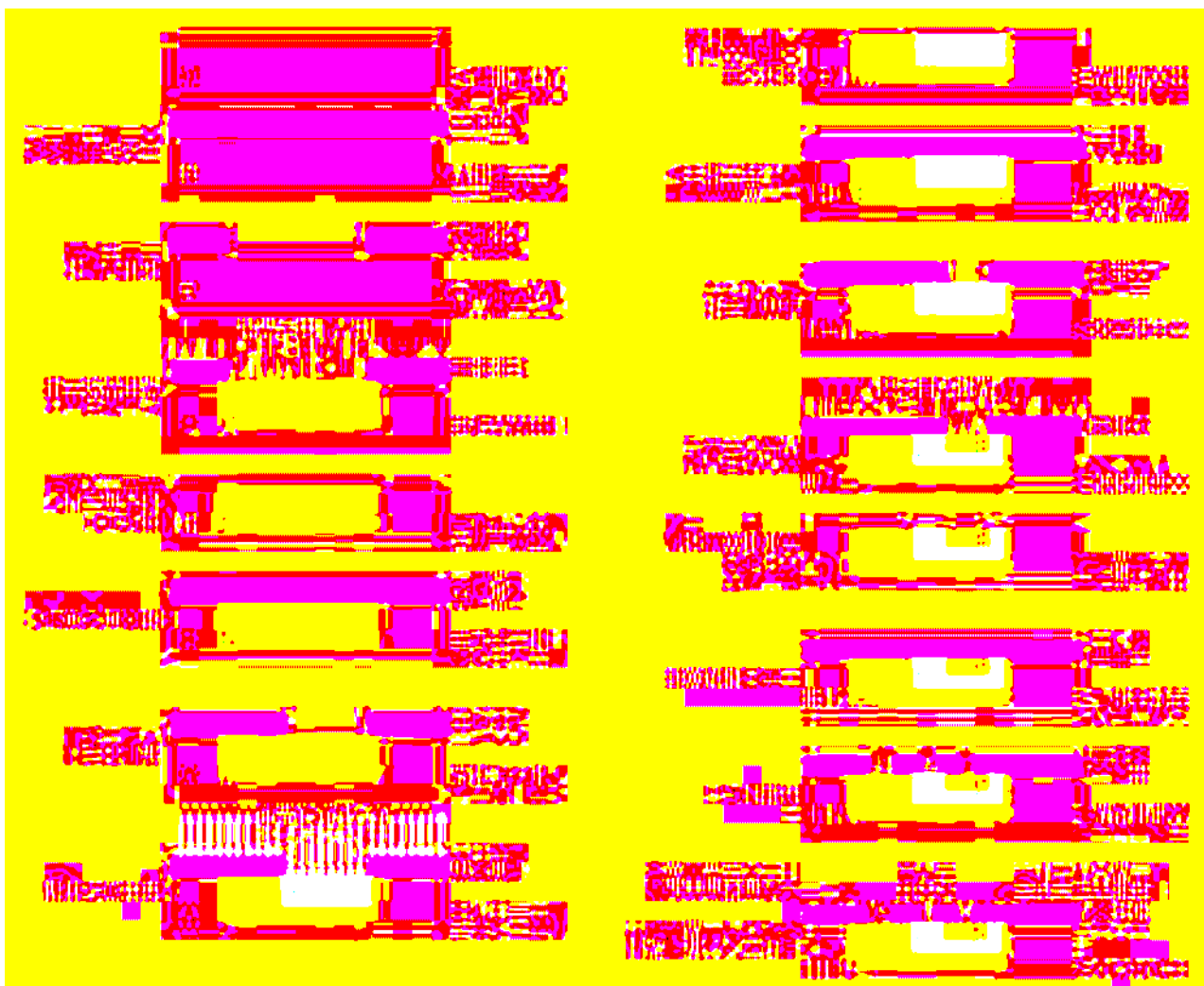



Figura 10 *Realizzazione di un transistor pnp*

sitivi "trough hole" quando per il montaggio su circuito stampato è necessario avere un foro passante, mentre si parla di componenti SMD quando il montaggio è di tipo superficiale (la saldatura viene fatta sul lato componenti). Ciascun tipo di package è identificato con un acronimo che ne riassume le caratteristiche. La figura 14 riporta alcuni tipi di package con i relativi acronimi identificativi. Ovviamente spesso uno stesso dispositivo viene incapsulato in diversi tipi di package in modo da poter soddisfare tutte le esigenze di progettazione.

APPROFONDIMENTI

Per chi volesse approfondire l'argomento ecco un elenco di risorse bibliografiche:

- R.S. Muller, T.I. Kamins:
  "Dispositivi elettronici nei circuiti integrati", Bollati Boringhieri, 1993.
- J.M. Rabaey:
  "Digital Integrated Circuits, a Design Perspective", Prentice Hall, 1996.
- N. Weste, K. Eshra Ghian:
  "Principles of CMOS VLSI Design, a System Perspective", Addison Wesley, 1992.
- S.M. Sze:
  "Vlsi Technology", McGraw-Hill, 1988.
- G. Soncini:
  "Tecnologie Microelettroniche", Boringhieri, 1986.

Per lavorare nel campo della progettazione dei circuiti integrati è necessaria una laurea in ingegneria elettronica con specializzazione in Microelettronica, oppure una laurea in fisica.

Figura 11 Un granello di polvere su una pista da 1μm

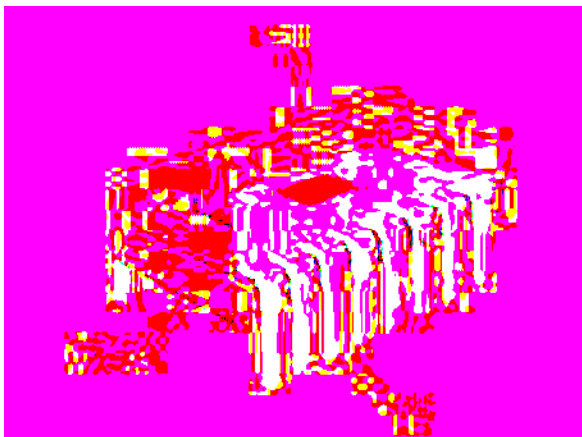



Figura 12 Il circuito integrato completo di package

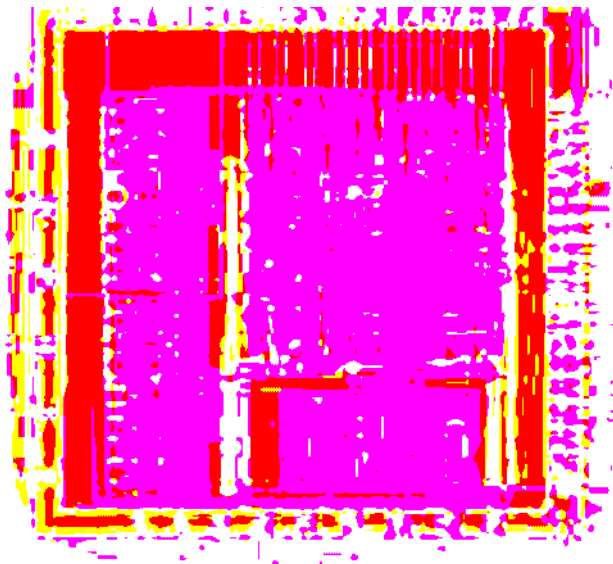



Figura 13 Un chip collocato correttamente nel proprio package

Figura 14 Alcuni comuni tipi di package per circuiti integrati

Alimentatori switching:

Torniamo, in questa puntata, a presentare un circuito pratico, ma sfrutteremo l'occasione per studiare un'ulteriore tipologia di switching, molto importante ed utilizzata: la flyback. Per la parte teorica accenneremo all'importante concetto dell'isolamento ingresso-uscita, fondamentale introduzione alle ultime due puntate del corso.

L'ISOLAMENTO INGRESSO/USCITA

Tutti siamo abituati a convivere con dispositivi elettrici ed elettronici che "prelevano" energia dalla rete elettrica. Sapete anche che, in generale, nelle case italiane è presente una tensione alternata sinusoidale con valore efficace di circa 230V (tolleranza ±10%), alla frequenza di 50Hz. Questa tensione viene comunemente denominata "**monofase**", in quanto **uno solo** dei due fili di alimentazione, ha un potenziale diverso rispetto alla terra (in situazioni normali), mentre l'altro viene messo a terra presso la centrale di distribuzione, vedi figura 1. Questo sistema di distribuzione viene denominato TT (Terra-Terra), chiedo venia agli elettrotecnici se questa spiegazione dovesse risultare lacunosa e/o approssimativa.

Il filo con potenziale sollevato rispetto alla terra, viene indicato normalmente come "fase", LIVE in inglese ed abbreviato con la lettera "L", mentre l'altro è il cosiddetto "neutro", NEUTRAL in inglese, abbreviato in genere con la lettera "N". Con la lettera "E" viene individuata invece la terra, EARTH in inglese.

Notate che in Italia le prese di corrente monofase non sono polarizzate, cioè non è possibile a vista determinare quale terminale è il neutro e quale è la fase (se non erro in Inghilterra le prese sono invece polarizzate).

Allo scopo sono però adatti i cosiddetti cacciavite "cercafase", che contengono al loro interno una piccola lampadina al neon, che si accende per mezzo di una debolissima ed innocua corrente tra la "fase" e la terra, attraverso il nostro corpo. Ancora meglio si può utilizzare un tester in posizione Vac e misurare le tensioni tra ciascuno dei due poli e la terra: solo in un caso dovreste leggere circa 230V. Se anche nell'altro caso si dovesse leggere una tensione superiore a pochi volt, avete un impianto elettrico difettoso (oppure siete tra i pochissimi che hanno ancora la rete a due fasi).

Se accidentalmente il nostro corpo andasse in contatto tra la fase e terra, la corrente che vi scorrerà, attraversandolo, potrebbe risultare molto pericolosa, vedi figura 2.

Figura 1 *Schema di distribuzione monofase*

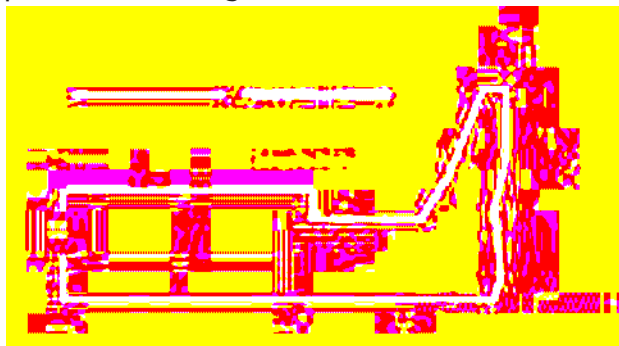
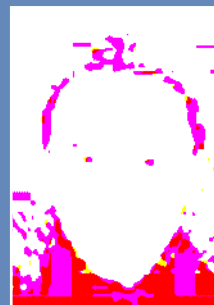



Figura 2 *Shock elettrico per contatto tra fase e terra*

Switching flyback e l'isolamento ingresso-uscita

di Romano Bernarducci
r.bernarducci@farelettronica.com

Per evitare questo, in tutti gli impianti elettrici civili moderni, è obbligatorio l'inserimento del cosiddetto dispositivo "salvavita" o interruttore differenziale. In pratica si tratta di un dispositivo che confronta le correnti che scorrono nei due fili della rete, fase e neutro, e che interrompe la fornitura di energia elettrica se dovessero essere diverse. Questo è quanto accade in figura 2, dove si vede chiaramente come la corrente scorra nel filo di fase (L), ma non in quello neutro (N).

In quasi tutti gli apparati elettronici, in genere la tensione 230V alternata non viene utilizzata direttamente, ma opportunamente trasformata, ridotta e convertita in una o più tensioni continue.
Questa conversione può avvenire tramite un trasformatore di rete a 50Hz, oppure per mezzo di uno switching collegato direttamente alla rete, e perciò denominato in inglese, *off-line* (che non significa "spento", ma appunto "collegato direttamente alla rete").
Supponiamo di avere un TV color, un DVD, un videoregistratore o quant'altro, con una presa accessibile dall'utente: se non ci fosse l'isolamento galvanico tra la rete elettrica e la massa della presa, toccando contemporaneamente quest'ultima e la terra potremmo trovarci nella situazione di shock elettrico di figura 2. Per questo motivo, nella stragrande maggioranza dei dispositivi elettronici, **la massa elettrica circuitale non è MAI collegata galvanicamente alla rete, ma è da questa isolata per mezzo di opportuni dispositivi.**

Una curiosità: tempo fa ebbi la stolta idea di aggiungere una presa cuffia ad un TV color 14 pollici, di una nota marca giapponese, che ne era sprovvista. Smontato il TV, eseguito un piccolo forellino sul contenitore plastico per fissare il jack (metallico), collegati i fili... suono in cuffia ok, peccato che ogni volta che toccavo il jack, avvertissi una (fortunatamente) piccola scossa. Solo a quel punto ho verificato lo schema elettrico ed ho notato con orrore che la massa elettrica di tutto il TV era collegata direttamente alla rete! Forse aveva ragione il costruttore, in quanto nessuna parte metallica era, di fabbrica, accessibile dall'utente, non avendo il TV neanche la presa SCART ed avendo provveduto a sprofondare nella plastica anche la massa della presa antenna! Mi auguro tuttavia che queste barbare "usanze" siano ormai scomparse, ma... fate attenzione, sempre e comunque, prima di mettere le mani su un circuito alimentato dalla rete che non conoscete!

Possiamo concludere il paragrafo affermando che:

> Un alimentatore da rete, oltre alla trasformazione / condizionamento della tensione, deve fornire sempre anche l'ISOLAMENTO GALVANICO.

NOTE SUGLI AVVOLGIMENTI

Qualche lettore mi ha chiesto delucidazioni in proposito ai "pallini" comunemente utilizzati nella simbologia del trasformatore. Il "pallino" altro non è che un **riferimento** identico per tutti i diversi avvolgimenti, che personalmente io considero essere l'INIZIO dell'avvolgimento.
Ovviamente le cose non cambiano se considererete il "pallino" la FINE dell'avvolgimento, purché questa scelta valga per tutti gli avvolgimenti. In entrambi i casi **il verso di realizzazione di tutti**

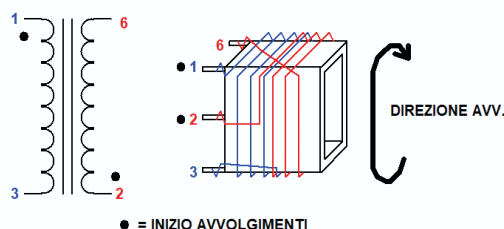



Figura 3 *Direzione degli avvolgimenti per un trasformatore generico*

gli avvolgimenti deve essere concorde, ad esempio sempre antiorario o sempre orario. Date uno sguardo alla figura 3 per ulteriori delucidazioni. I numeri 1, 3, 2, 6 sono quelli ipotetici di un supporto di bobina a 6 poli totali (non tutti utilizzati).

LA TIPOLOGIA "FLYBACK"

Questa tipologia è usata moltissimo in campo consumer, grazie alle sue caratteristiche di semplicità, quindi economicità ed affidabilità. Lo schema elettrico di principio è riportato in figura 4.
Ovviamente, come interruttore 1 potete utilizzare anche un transistor bipolare NPN, oppure anche un PMOS collegato tra VIN+ e il "pallino" del primario, collegando l'altro capo del trasformatore a massa. Analogamente, anche se in figura è stato utilizzato un diodo schottky come interruttore 2, nei casi appropriati questo potrà o anzi dovrà essere di tipo standard veloce.
Si nota immediatamente che, quando l'interruttore 1 è attivo, la tensione di ingresso viene applicata ai capi del primario del trasformatore, con il polo positivo coincidente con il "pallino". Per le note regole sui trasformatori, la tensione di ingresso, moltiplicata per il rapporto Ns/Np, sarà presente anche sul secondario, sempre con il polo positivo coincidente con il "pallino". Dalla figura 4 è però immediato vedere che il diodo 2, con la polarità indicata, è polarizzato inversamente e quindi non fa passare alcuna corrente.
Questa è la caratteristica fondamentale della tipologia *flyback*, che lo differenzia radicalmente dalla *forward*: **durante il periodo di ON, nessuna corrente scorre dall'alimentatore di ingresso verso il carico**. Piuttosto questa corrente, che ovviamente sale a rampa poiché scorre in un'induttanza (il primario del trasformatore), viene convertita in energia magnetica ed immagazzinata nel nucleo del trasformatore.
A differenza della tipologia *forward*, in cui il trasformatore è un mezzo di trasporto dell'energia **in tempo reale**, nella tipologia *flyback* il trasformatore ha la funzione di un vero e proprio serbatoio, che **prima** viene riempito (fase ON) e solo **dopo** viene svuotato (fase OFF).
I diagrammi della figura 5 chiariscono meglio il concetto.
Dal grafico verde di figura 5, notate che ho disegnato la curva Is (corrente nell'avvolgimento secondario) in modo che la corrente scenda a zero PRIMA dell'inizio del nuovo ciclo di ON. Come immaginerete, questa modalità di funzionamento, **che si impone attraverso la scelta del trasformatore e della frequenza di funzionamento, prende il nome di modalità DISCONTINUA,** abbreviato a volte come DCM. Nel caso che invece la corrente non riesca a scendere a zero prima dell'inzio del ciclo ON successivo, si parla di modalità CONTINUA o CCM, e i diagrammi delle correnti Ip e Is si modificano con l'aggiunta di uno scalino. Abbiamo già parlato precedentemente, durante la puntata di Marzo 2005 (FE237), dei vantaggi e degli svantaggi di ciascuna modalità: ricordo qui che, in generale, **la modalità DM fornisce una risposta ai transitori più rapida.**
Il prezzo che si paga è quello di una maggiore corrente di picco, a parità di caratteristiche, rispetto alla CM.
Nella maggioranza dei casi, **la *tipologia* flyback viene impiegata in modalità DCM,** ed è questa quindi che andremo ad analizzare ed utilizzare.

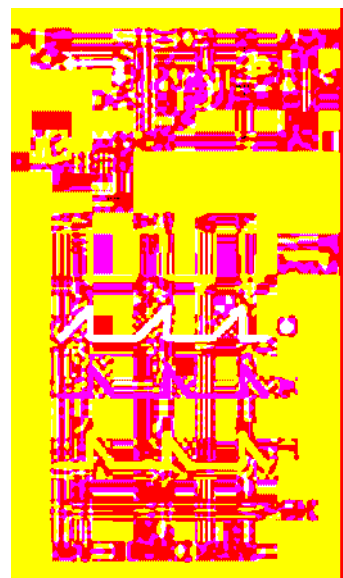



Figura 5 Forme d'onda caratteristiche per tipologia "flyback"

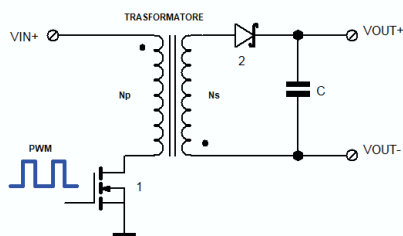



Figura 4 Schema di principio per tipologia "flyback"

TIPOLOGIA "FLYBACK": PARAMETRI DI PROGETTO

Supponiamo di voler realizzare un alimentatore switching, di tipo "*flyback*", avente le seguenti caratteristiche:

$V_{IN,NOM}$ = 24VDC ±25%, cioè:
$V_{IN,MIN}$ = 18VDC

$V_{IN,MAX}$ = 30VDC

V_{OUT} = 5VDC
$I_{OUT,MAX}$ = 1A

TIPOLOGIA "FLYBACK": CALCOLI PRELIMINARI

1. Calcoliamo la potenza di uscita massima:
   $P_{OUT,MAX} = V_{OUT} * I_{OUT,MAX} = 5 * 1 = 5W$

2. Ipotizziamo l'efficienza η pari al 75% e calcoliamo la potenza massima assorbita dall'ingresso:
   $P_{IN,MAX} = P_{OUT,MAX} / \eta = 5 / 0,75 = 6,667W$

3. Ad ogni ciclo ON, l'energia immagazzinata nel trasformatore (cioè nell'induttanza del primario) si calcola con la:
   $E_L = 0,5 * Lp * (I_{p,MAX})^2$

4. La **totalità** di questa energia (**dobbiamo imporre il funzionamento discontinuo**) viene trasferita al carico, per mezzo dell'avvolgimento secondario, ad ogni ciclo. Quindi la potenza trasferita, pari all'energia erogata in un secondo, è data da $E_L * f_{SW}$ (f_{SW} è la frequenza di switching) e deve essere uguale alla $P_{IN,MAX}$ calcolata al passo 2:
   $P_{IN,MAX} = E_L * f_{SW} = 0,5 * Lp * (I_{p,MAX})^2 * f_{SW}$

5. Consideriamo ora che la $I_{p,MAX}$ dipende a sua volta (oltre che dall'induttanza del primario L_p), anche dalla tensione di ingresso e dal tempo di ON, secondo la nota formula:
   $I_{p,MAX} = (V_{IN} – V_{DROP}) / Lp * T_{ON}$
   (V_{DROP} è la caduta totale di tensione sul MOS e sulle resistenze)

6. Dai passi 4 e 5 precedenti, si ottiene con semplici manipolazioni algebriche:
   $P_{IN,MAX} = 0,5 * (V_{IN} – V_{DROP})^2 * T_{ON}^2 * f_{SW} / Lp$

Nel caso peggiore, vale ovviamente la $V_{IN} = V_{IN,MIN}$. Ipotizziamo poi V_{DROP} = 1V per le cadute resistive; come al solito, se alla fine questa ipotesi dovesse risultare troppo distante dalla realtà, sarà possibile tornare a questo passo sostituendo il valore più corretto.
Nel caso di utilizzo di tipologia di controllo Current Mode, come è il caso in questo esempio, è bene che il duty-cycle non superi il 50%. In caso contrario sappiamo che sarebbero richiesti componenti aggiuntivi per effettuare la cosiddetta "slope compensation". Per sicurezza utilizziamo il valore massimo del 45% per il duty-cycle, quindi $T_{ON} = 0,45 * (1 / f_{SW})$, e sostituendo otteniamo:

$P_{IN,MAX} = 0,5 * (18 – 1)^2 * 0,45^2 * (1 /f_{SW})^2 * f_{SW} / Lp$

Questo valore deve essere maggiore o almeno uguale al valore desiderato, cioè 6,667W.
Effettuando i calcoli e semplificando, si ottiene la:

$$Lp * f_{SW} <= 4,389$$

Resta ancora una volta dimostrato che esistono, in teoria, infinite coppie Induttanza/Frequenza che soddisfano i criteri desiderati per un alimentatore switching.
Tabuliamo i risultati per varie frequenze, ci serviranno nel momento della selezione del nucleo.

TIPOLOGIA "FLYBACK": SCELTA DEL NUCLEO

Dalla Tabella presentata in figura 6 della puntata 7, notiamo che anche un piccolo nucleo E20 (20x20x6 mm) riesce a fornire ben 26W a 25kHz, anche se in configurazione push-pull. La configurazione flyback utilizza il nucleo magnetico in un solo quadrante B-H (**la corrente nel primario scorre sempre in una sola direzione**), a differenza dei due quadranti sfruttati dalla push-pull, e quindi l'energia disponibile sarà minore e circa pari a **0,5-0,7 volte** quella della tabella appena citata. Nonostante questo, il nucleo E20/10/6 è

f_{SW} [kHz]	Lp <= di [μH]
20	219,45
30	146,30
40	109,73
50	87,78
60	73,15
70	62,70
80	54,86
90	48,77
100	43,89

Tabella 1 Coppie frequenza/induttanza

chiaramente sovradimensionato per il nostro switching, ma è facile da reperire e le "ampie" dimensioni faciliteranno la realizzazione degli avvolgimenti. Come esercizio lascio a voi la progettazione di switching ben più potenti, con lo stesso nucleo. Scelto il nucleo, restano da definire altri due aspetti:

- Il tipo di materiale, che dipende strettamente dalla frequenza operativa.
- Il "gap", cioè lo spazio vuoto esistente tra la gambe centrali di molti nuclei in ferrite, vedi figura 6.

Cosa succede al nostro nucleo con l'introduzione del GAP?
Considerate che **l'aria non è un materiale ferromagnetico**, quindi, in generale (mediamente) la capacità di allineamento delle isole magnetiche diminuirà. La diminuzione della permeabilità del nucleo, implica che per ottenere la stessa induttanza, saranno necessarie più spire nell'avvolgimento, ma questo di per sé non è solo uno svantaggio, infatti:

L'aggiunta del "gap" in un nucleo magnetico, aumenta la sopportazione di corrente, cioè la corrente massima che è possibile fare scorrere sull'avvolgimento prima del raggiungimento della saturazione.

Questa caratteristica è di fondamentale importanza in tutti i casi in cui l'induttanza o il trasformatore sono attraversate da grandi correnti. Vedremo che questo è il caso della tipologia flyback, in modalità discontinua.
Per il nostro switching, funzionante a frequenza sicuramente minore di 100kHz, sceglieremo un nucleo in materiale standard, quale l'N27 della EPCOS o equivalenti di altre case.
La scelta del "gap" necessario richiede qualche calcolo in più. Infatti notate che il nucleo E20/10/6 viene fornito dalla Epcos in 5 differenti versioni:

- Senza gap (g = 0)
- Con gap da 0,09mm
- Con gap da 0,17mm
- Con gap da 0,25mm
- Con gap da 0,50mm

Come calcolare l'effetto del gap sull'induttanza? Per rispondere a questo introduciamo un nuovo (semplice!) concetto.

IL NUMERO "MAGICO" A_L

Una richiesta molto comune che mi è giunta già diverse volte da voi lettori, è la seguente:

Come calcolare il numero di spire necessarie per ottenere un dato valore di induttanza, con un certo nucleo magnetico?

La risposta teorica sarebbe abbastanza complicata, in quanto deve tener conto di diversi fattori come la forma del nucleo, le sue dimensioni, il tipo di materiale magnetico, eccetera.
Per fortuna i costruttori di ferriti ci sono venuti incontro:

Per ogni nucleo e materiale magnetico, esiste un valore A_L che, moltiplicato per il numero delle spire al quadrato, fornisce l'induttanza.

Matematicamente:

$$L = A_L * N^2$$

Il valore A_L, in genere in nanoHenry (nH, 10^{-9} Henry), è riportato nei datasheet.
Per fare un esempio, il nucleo E20/10/6, senza GAP, di materiale N27, presenta un valore di A_L uguale a 1300nH, con ampia tolleranza (+30% -20%).
Se con questo nucleo volessimo realizzare un'induttanza del valore di 100µH, occorrerebbero quindi, invertendo la formula precedente:

$N = (L / A_L)^{0,5} = (100e\text{-}6 / 1300e\text{-}9)^{0,5} = 8,77$ spire, arrotondate a 9

La tabella 2 riporta i valori di A_L per tutti i tipi

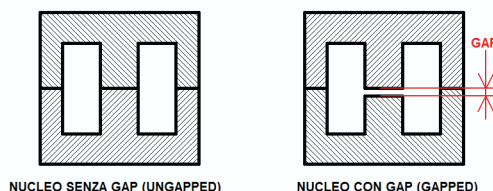



Figura 6 *Nucleo ad "E" senza "gap" e con "gap"*

di nucleo E20/10/6, materiale N27, ed i rispettivi codici.
A questo punto riprendiamo la tabella 1, riportante le coppie frequenza/induttanza da 20kHz a 100kHz, e completiamola con il numero di spire necessario ad ottenere tale induttanza, per i vari nuclei. Otteniamo la tabella 3.

Anche se il tutto sembra complicato, un banalissimo spreadsheet tipo OpenOffice o Excel può semplificarvi molto la vita (sapendolo usare, naturalmente).

Forza, ci siamo quasi... domanda: **secondo voi, dalla tabella 3, qualunque tripletta frequenza/nucleo/numero spire va bene?**

Se avete risposto si, avete sbagliato... ricordate quanto vi ho detto più volte:

In uno switching il nucleo magnetico NON deve mai raggiungere la saturazione, cioè B deve sempre essere MINORE di B_{SAT}

Ricordando quanto detto nella puntata 7, è immediato calcolare la densità di flusso massima:

$$B_{MAX} = V_{IN} * T_{ON} / (N * Amin) <= B_{SAT}$$

dove $V_{IN} = V_{IN,MIN} = 17V$, $T_{ON} = 0,45 * (1 / f_{SW})$, Amin = 31,6mm² per il nucleo E20. Supponiamo che B_{SAT} possa raggiungere al massimo 0,2 Tesla, cioè 200 mT. Riportiamo le densità di flusso calcolate, per ciascun nucleo/frequenza, in tabella 4.

Notate come il nucleo senza gap e quello con gap = 0,09mm, entrino in saturazione a qualunque frequenza!

Gap nucleo E20/10/6 (mat. E27) [mm]	A_L [nH]	Codice EPCOS
0,00	1300	B66311-G-X127
0,09	350	B66311-G90-X127
0,17	224	B66311-G170-X127
0,25	170	B66311-G250-X127
0,50	106	B66311-G500-X127

Tabella 2 Coppie gap/A_L

Il primo nucleo accettabile è invece quello con gap = 0,17mm, fatto funzionare alla frequenza di 80kHz.
Inutile dirvi che è quello che useremo nel nostro progetto (con 16 spire di avvolgimento primario, come da tabella 3).
Notate infine, dalle tabelle 3 e 4, che aumentando il

f_{SW} [kHz]	Lp <= di [μH]	Nucleo B66311-G-X127 A_L = 1300nH SPIRE	Nucleo B66311-G90-X127 A_L = 350nH SPIRE	Nucleo B66311-G170-X127 A_L = 224nH SPIRE	Nucleo B66311-G250-X127 A_L = 170nH SPIRE	Nucleo B66311-G500-X127 A_L = 106nH SPIRE
20	219,45	13	25	31	36	46
30	146,30	11	20	26	29	37
40	109,73	9	18	22	25	32
50	87,78	8	16	20	23	29
60	73,15	8	14	18	21	29
70	62,70	7	13	17	19	24
80	54,86	6	13	16	18	23
90	48,77	6	12	15	17	21
100	43,89	6	11	14	16	20

Tabella 3 Numero di spire necessarie per ogni nucleo e frequenza

gap a 0,25mm o 0,50mm, si può restare ancora nei limiti per la densità di flusso, ma a scapito di un maggior numero di spire a parità di frequenza --> maggiori perdite nel rame --> minore efficienza.

IL CIRCUITO

Questa volta ve l'ho fatto sudare ancora più del solito, e chiedo venia... ecco il circuito completo, già testato e funzionante.
Purtroppo per motivi di spazio sono costretto a chiudere qui, mi auguro che lo realizzerete e lo userete come spunto per progetti più ambiziosi.

CONCLUSIONI

Nella prossima puntata accennerò ai semplici calcoli relativi allo switching flyback appena presentato, ma soprattutto vedremo come "espandere" questo progetto in modo da ottenere con **estrema semplicità** più tensioni di uscita contemporaneamente. Non mancherà l'approfondimento delle problematiche di isolamento ingresso-uscita. A rileggerci fra 30 giorni!

f_{SW} [kHz]	Lp <= di [µH]	Nucleo B66311-G-X127 A_L = 1300nH B_{MAX} [mT]	Nucleo B66311-G90-X127 A_L = 350nH B_{MAX} [mT]	Nucleo B66311-G170-X127 A_L = 224nH B_{MAX} [mT]	Nucleo B66311-G250-X127 A_L = 170nH B_{MAX} [mT]	Nucleo B66311-G500-X127 A_L = 106nH B_{MAX} [mT]
20	219,45	932	438	387	337	266
30	146,30	716	395	316	275	217
40	109,73	659	342	273	238	188
50	87,78	589	306	245	213	168
60	73,15	538	297	223	195	154
70	62,70	498	258	207	180	142
80	54,86	466	242	193	168	133
90	48,77	439	228	182	159	125
100	43,89	417	216	173	151	119

Tabella 4 *Densità di flusso massima per ogni nucleo e frequenza*

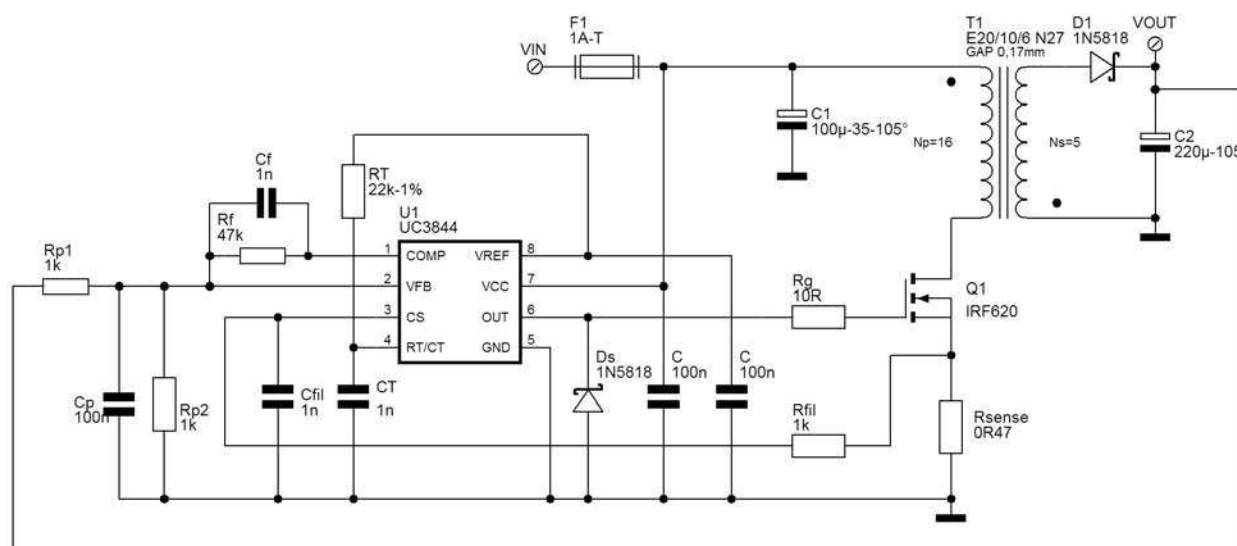



Figura 7 *Switching 5V-1A in tipologia flyback*

SISTEMI DI VIDEOSORVEGLIANZA

WIRELESS

Accensione automatica

Con questo circuito ed un semplice software si può programmare l'accensione di un PC sfruttando delle proprietà poco conosciute di una qualsiasi scheda madre (ATX) il WoR, WoL od il WoM. Il software consente anche l'avvio automatico di un programma all'accensione.

Le schede madri di recente produzione presentano diverse proprietà che sono poco sfruttate, tra queste esistono quelle definite eventi di Wake-Up e di cui in seguito si vedranno differenze e peculiarità tecniche. Come accenna il termine sono caratteristiche che consentono un controllo sull'avvio del computer che non dipende dalla comune pressione del tasto Power presente sulla parte frontale del Case. La possibilità di far partire un PC in nostra assenza non ha gran valore se non vi è associata un'azione che porti, ad accensione avvenuta, il computer ad eseguire uno determinato comportamento è quindi necessario un programma residente sul computer che si accorga di una accensione automatica e che agisca su ordini precedentemente definiti. Tanto per fare degli esempi d'utilizzo dopo un evento di wake-up il PC si avvia, lo si può fare scegliendo di forzare, automaticamente, una connessione ad internet ed in seguito lanciare un software di file sharing (Emule, Shareaza etc). Chi avesse una scheda satellitare potrebbe pensare di accendere il PC ad un'ora prestabilita, far partire un programma per la visualizzazione della TV satellitare che registri un programma d'interesse. L'hardware necessario per consentire una comunicazione adatta col PC e la gestione dei segnali WoL, WoR o WoM è modesta e sfrutta tre integrati molto comuni il PIC16F84 (di cui programmatore e documentazione tecnica sono facilmente reperibili sui precedenti numeri di Fare Elettronica), l'RS232 ed un NE555.

INTRODUZIONE A WOR, WOL E WOM

WoR, WoL e WoM sono rispettivamente acronimi di *Wake on Ring*, *Wake on LAN* e *Wake on Modem*, almeno una di queste funzionalità è implementata in una qualsiasi scheda madre ATX ma è possibile trovarne anche più di una contemporaneamente. Nascono per consentire al computer di "svegliarsi" (Wake), ossia di accendersi, obbedendo ad un segnale di controllo. Il WoL nasce con l'idea di consentire l'accensione del PC tramite una scheda LAN e quindi di rendere possibile la gestione di questo evento tramite un altro computer che è connesso alla stessa rete LAN. Il WoR nasce sull'idea di consentire un'accensione comandata da telefono, opportunamente connesso al PC, utile ad esempio nel caso esista la

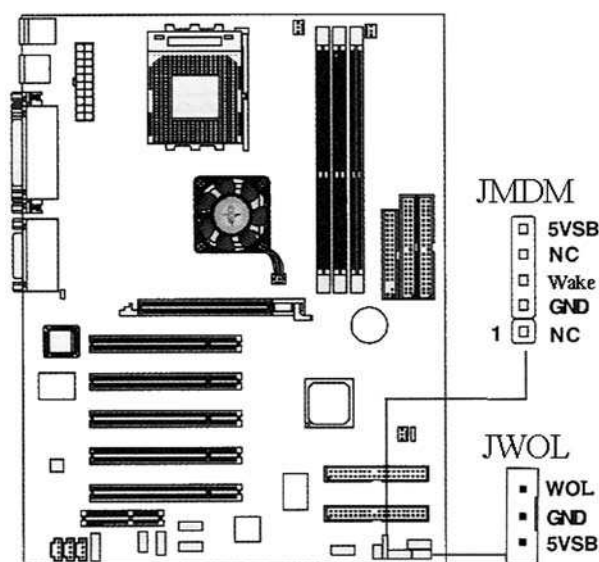



Figura 1 *Scheda madre e particolare sui connettori JMDM e JWOL*

del PC

di Lucio Del Veneziano
lukkiov@libero.it

necessità di ricevere un fax sul calcolatore. Anche il WoM strettamente legato al WoR consente l'accensione del computer via Modem.
Sulla scheda madre saranno presenti due connettori con le piedinature simili od identiche a quelle in figura 1, queste specifiche tecniche sono descritte nel manuale della scheda.

Il connettore indicato con JWOL, fornisce al pin 5VSB una tensione di 5V, che servirà ad alimentare il circuito realizzato, prendendola dall'alimentatore del PC, il GND fornisce il riferimento di massa mentre il piedino indicato con WoL attende il segnale che consentirà l'accensione del computer. Questo segnale deve essere un impulso positivo (segnale sempre a 0 che per un breve periodo si porta a 5V).
Le piedinature dei connettori relativi al WoR o al WoM sono identiche e contrassegnate in figura 1 con la sigla JMDM (in figura 1 è riportato il caso di una MainBoard che supporta il WoM e WoL). Le considerazione su Vsb, Gnd sono le stesse fatte per il JWoL mentre per ciò che riguarda i segnali bisogna dire che sono in generale impulsi negativi nel caso del WoM ed impulsi positivi nel caso di WoR (un impulso negativo si intende un segnale sempre a 5V e per un breve periodo passa a 0V).

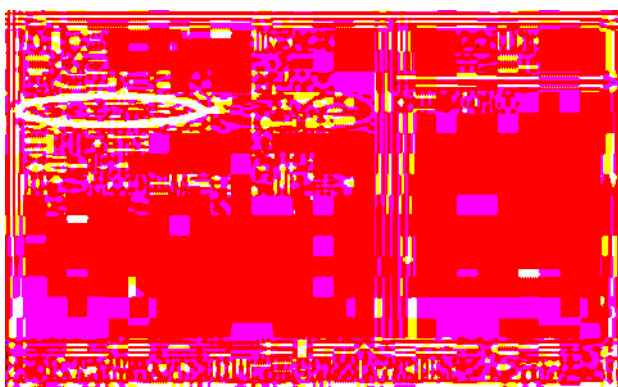



Figura 2 Settaggio BioS

Infine questi eventi di accensione vanno in generale abilitati via Bios, è consigliabile sfogliare il manuale utente della scheda madre in modo da ottenere una corretta impostazione.
In figura 2 un tipico esempio di schermata Bios aperta sulla pagina "Power management" dove si può gestire l'attivazione degli eventi di Wake Up impostandoli Enabled/Disabled come cerchiato in blu in figura.

SCHEMA ELETTRICO

Lo schema elettrico è rappresentato in figura 3. La tensione di 5V continui necessaria ad alimentare il circuito viene prelevata dal connettore dalla scheda madre (in figura 1 il connettore JMDM o JWOL) e portata al piedino 5VSB del connettore J1 tramite un cavo appropriato di cui si discuterà in seguito. Stesso discorso per la massa che appare come GND sia sulla motherboard che in J1 e che in più sarà anche il riferimento di tensione della porta seriale collegata al piedino 5 di J2.
Il terzo pin del connettore J1 è l'uscita che gestisce l'evento WoR sulla scheda madre, viene controllato dal pin RB3 del uC che lo mantiene sempre a 0V tranne quando si deve accendere il PC che si porta a 5V.
La connessione di Q2 con WoR è del tipo a "collettore aperto" ossia se Q2 è "spento" (quando le tensioni su RB3 ed alla base di Q2 valgono zero) il WoR è scollegato e non succede nulla, se però Q2 si "accende" (quando la tensione in RB3 è 5V e quindi quella di base di Q2 si porta ad almeno 0,7V) la tensione di WoR si porta a 0V determinando l'accensine del PC.
L'ultimo pin del connettore J1 è l'uscita che gestisce l'evento WoL sulla scheda madre questo è però regolato da un impulso positivo, opposto cioè al caso precedente del WoR. Viene preso direttamente dall'uscita RB3 del PIC.

La porta RS232 del PC deve inviare e ricevere

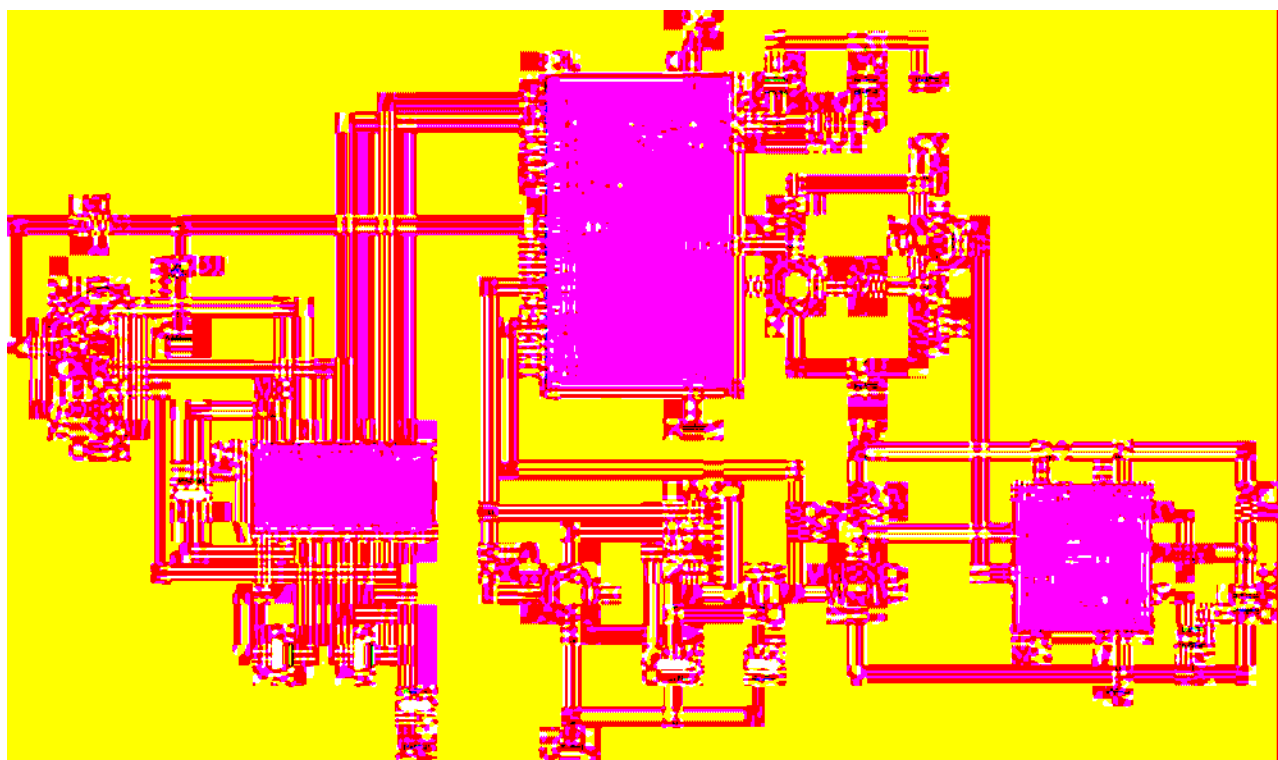



Figura 3 Schema elettrico

dati dal circuito ma lo standard di comunicazione di questo protocollo utilizza due valori digitali diversi da quelli che usa il microcontrollore, il primo usa -10V e +10V ed il secondo +5V e 0V, è necessario quindi un integrato che traduca questi due linguaggi diversi: il Max232. Questo necessita di solo quattro componenti esterni per poter funzionare ossia quattro condensatori elettrolitici. Il Max232 riceve dati da RxD (pin 3 di J2) della seriale e li invia al RA1 del PIC16F84 così come il PIC invia da RA0 verso il piedino 10 del Max232 e questo reinterpreta ed invia sequenze di impulsi alla seriale (pin 2 di J2). Tramite R3 e il diodo Zener D1 possiamo controllare l'interrupt del PIC, che ci permette di interrompere il microcontrollore dal conteggio, semplicemente agendo via software sul RTS (pin 8 di J2) della seriale. Lo zener porta la tensione di RTS che può essere +/- 10V ai 0/5V su RB0.
Il PIC verrà programmato in modo da utilizzare il quarzo a 4MHz presente tra i pin OSC1/CLKIN e OSC2/CLKOUT.
Il blocco rimanente composto dall'NE555 ed i transistor Q1, Q3 serve per gestire al meglio la fase di reset del microcontrollore. Il pin RB5 fornisce un breve impulso negativo all'NE555, questo si preoccuperà di prolungare la durata dell'impulso e di renderlo positivo, ciò è reso possibile dal fatto che il 555 è utilizzato in configurazione monostabile. Infine dal pin 3 di uscita il 555 invia l'impulso risultante alla base di Q1. L'utilizzo di Q1 e Q3 rende compatibile l'uscita dell'NE555 (pin 3) con il pin MCLR del PIC, infatti in uscita il 555 fornisce impulsi

1° - BC	2° - BD1	3° - BD2	4° - BD3	5° - BID1	6 - BID2
Byte di controllo per evitare errori di trasmissione vale sempre 'B1'	Primo Byte di dati che rappresentano i secondi (1A nell'esempio sopra)	Secondo Byte di dati che rappresentano i secondi (EA nell'esempio sopra)	Terzo Byte di dati che rappresentano i secondi (A0 nell'esempio sopra)	Primo Byte ID che rappresenta il numero della programmazione	Secondo Byte ID che rappresenta il numero della programmazione

Tabella 1 Stringa 6 Byte inviata dal PC verso il Pic16F84

Dal 15 Novembre

in tutte le Edicole

OMAGGIO per gli ABBONATI*

* Se sei un nuovo abbonato devi abbonarti entro il 30 Settembre 2005.
Se il tuo abbonamento scade entro Novembre 2005 devi rinnovarlo entro il 30 Settembre 2005.

Programma Principale

Inizializzazione Porte I/O

Lettura Byte di controllo

Ricevuto 'B1' ?

No

Si

Leggo 1°,2° e 3° Byte del secondi e scrivo in Sec_low, sec_mid e sec_high

Leggo 1° e 2° della ID

Scrivo i 5 Byte nella EEPROM

Invio questi 5 Byte al PC per conferma, se ci sono errori il PC allerta con messaggio

sec_1 = sec_low+1
sec_m = sec_mid+1
sec_h = sec_high+1

Aspetto 1 secondo

Sec_l = sec_l -1

Sec_low = sec_low -1

Sec_l = 0 ?

No

Si

Sec_m = sec_m -1

Sec_mid = sec_mid -1

Sec_m = 0 ?

No

Si

Sec_h = sec_h -1

Sec_hig = sec_hig -1

Sec_h = 0 ?

No

Si

Impulso su RB3 di 1 s

Cancello EEPROM

Sleep

Figura 4 *Diagramma di flusso del programma principale*

positivi, mentre per il PIC la tensione su MCLR deve essere sempre positiva e si deve portare a 0V nel momento in cui si vuole resettarlo (impulso negativo), si deve quindi invertire l'uscita al pin 3 del NE555 e lo si fa con Q3 che è in configurato come emettitore comune.
Mentre Q1 in configurazione collettore comune funziona da buffer tra appunto il Timer e la sezione invertente.

PRINCIPIO DI FUNZIONAMENTO

Tramite il software "kalendarjo.exe" si deve impostare la data e l'ora di accensione automatica del Personal Computer, questo periodo di tempo viene convertito in secondi e rappresentato in notazione esadecimale. Ad esempio si effettua una programmazione il giorno 3/2/2005 alle 10:00 per accendere il PC il giorno 23/2/2005 alle 20:00, i secondi in questo periodo di tempo sono 1.764.000 che in notazione esadecimale valgono *1AEAA0* ossia tre Byte *1A*, *EA* e *A0*.

Una volta confermata la data di programmazione, il software invia un comando di interrupt al PIC e di seguito un comando che lo obbliga a resettarsi, adesso il PIC16F84A è in attesa di istruzioni. Le informazioni che adesso devono arrivare al PIC sono una stringa di 6 Byte descritta nella tabella 1.

Nel caso in cui il microcontrollore sia in attesa ed il primo byte che riceve non è quello di controllo (*B1*) capisce che c'è un errore e si porta in sleep mode ossia attende di essere resettato e lo fa in modalità a basso consumo. Se invece riceve B1 prosegue nell'attesa degli altri Byte che in sequenza sono BD1, BD2 e BD3, gli stessi di cui si discuteva in precedenza, 1A EA A0, che rappresentano i secondi.
Il massimo numero rappresentabile da 3 Byte è FFFFFF (che in binario è un numero a 24 bit) ed in decimale è pari 2^{24} = 16.777.215 secondi che risultano essere 194 giorni, 4 ore, 20 minuti e 15 secondi, questo è il massimo periodo di pro-

grammazione. Gli ultimi due Byte BID1 e BID2 identificano con un numero la programmazione ed è un dato utile al software su PC per far partire in automatico i programmi preselezionati ad un avvio del computer avvenuto con un evento di Wake-Up. In seguito il PIC inizia il conteggio alla rovescia decrementando, ad ogni secondo che passa, il numero, formato da tre Byte, che rappresenta i secondi mancanti all'evento. Quando i Byte si portano tutti e tre a 0 (ossia 000000) il conteggio termina e l'uscita RB3 viene portata a +5V per un secondo quindi il segnale WoL sarà un impulso positivo di un secondo mentre il WoR verrà portato a 0V per un secondo. L'accensione è avvenuta ed il PIC si porta in modalità a basso consumo nell'attesa di nuovi comandi. Da qui in poi gestisce tutto il software *Kalendarjo* che allo start up del computer controlla l'ID del PIC e se il conteggio è terminato, fa partire il programma che si era scelto di avviare in automatico.

DESCRIZIONE FIRMWARE DEL PIC16F84

Lo studio del firmware è condensato in due parti, il programma principale e la gestione dell'interrupt.

Programma principale

Il diagramma di flusso del primo è disegnato in figura 4.

Vengono da principio inizializzate le porte del PIC che si utilizzano, quindi si passa ad attendere il primo Byte da PC che serve come Byte di controllo se si ottiene il valore esadecimale B1 allora significa che il PC sta effettivamente tentando di trasmettere qualcosa al microcontrollore. Può infatti capitare, ad esempio durante una normale accensione del computer, che le uscite della porta RS232 cambino stato diverse volte e senza un preciso ordine se quindi non ci fosse il Byte di controllo il PC potrebbe portare ad una programmazione non voluta del PIC quando questo è in attesa di informazioni. Dopo il Byte di controllo il PIC attende e riceve i 3 Byte relativi ai secondi ed i 2 relativi all'ID che tramite una opportuna funzione vengono scritti sulla EEPROM e di seguito inviati al PC. Il software gestisce la ricezione di questi 5 Byte e la compara con quelli che aveva inviato lui, se sono uguali è tutto a posto altrimenti fa partire un messaggio di errore, questo processo serve a controllare se la ricezione dei secondi al PIC è avvenuta correttamente.

A questo punto si inizializzano tre variabili (sec_l, sec_m, sec_h) ponendole uguali alle variabili sec_low, sec_mid e sec_high a cui viene sommato 1, potrebbe sembrare ridondante definire tre nuove variabili ma è necessario procedere in questo modo perché, nel caso si verifichi un interrupt durante il conteggio alla rovescia, se ci si basasse sulle variabili sec_low, sec_mid e sec_high otterrei valori dei 3 Byte riferiti ai secondi errati. A questo punto inizia il ciclo di conteggio alla rovescia, si decrementano i Byte partendo da sec_l quando questo è 00 si decrementa sec_m se questo non si porta a 00 allora si riparte dall'inizio con sec_l = FF. Quando anche anche sec_m è 00 si passa a decrementare sec_h, se questo non vale 00 si riparte dall'inizio con sel_l = FF e sec_m = FF. Il conteggio termina quando tutte e tre le variabili sono azzerate. A questo punto viene creato

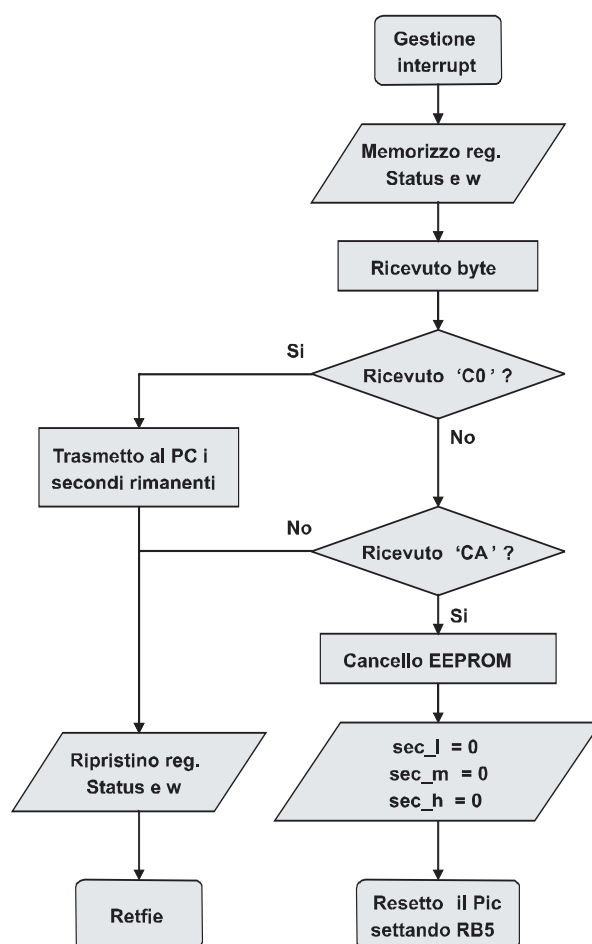



Figura 5 *Diagramma di flusso del programma di gestione interrupt*

Power core Flex

CONNETTI LE TUE IDEE CON LE NOSTRE

piccoli componenti, per grandi progetti: i tuoi.

Comprel Componenti Competenti

l'impulso positivo sulla porta RB3 dopodichè il PIC16F84 entra in modalità a basso consumo SLEEP in attesa di nuove istruzioni.

Gestione degli interrupt

Per interagire con il PC bisogna obbligare il microcontrollore ad interrompere le operazioni che sta eseguendo e lo si fa andando a settare la porta RB0 (pin 6), questo impulso fa partire un interrupt con conseguente esecuzione del programma schematizzato nel diagramma a blocchi di figura 5.

Vengono inizialmente salvati alcuni registri che saranno ripristinati al termine del ciclo della sequenza d'interrupt, quindi il PIC si mette in attesa di un comando da ricevere. Nel caso in cui non si presenti alcun dato in ingresso al PIC entro circa 400ms viene terminata la funzione relativa all'interrupt, vengono ripristinati i registri e si esce dall'interruzione con *retfie* riportandosi nel punto in cui il PIC era stato interrotto.
Se invece viene ricevuto qualcosa questo deve essere interpretato come due possibili tipi di comandi C0 e CA, con C0 si intende *controllo*.
Questo viene inviato dal PC al microcontrollore quando si vuole solo osservare quanto tempo manca all'evento di accensione, se quindi il comando ricevuto è questo allora il PIC risponde inviando i tre Byte relativi ai secondi ed identificati dalle variabili sec_low, sec_middle e sec_high quindi ripristina i registri ed esce con retfie.
Nel caso in cui il comando ricevuto è CA, che sta per *cancella*, significa che dal PC è arrivata una richiesta di cancellazione dell'evento di accensione quindi si cancella inizialmente la memoria EEPROM poi le variabili relative ai secondi (sec_low, sec_mid,sec_high, sec_l, sec_m, sec_h) ed infine si resetta il PIC settando la porta RB5 del PIC stesso.
In questo caso quindi il PIC dopo l'interrupt riparte dall'inizio del programma principale.
Se il comando ricevuto non è ne C0 ne CA, probabilmente c'è stato un errore di comunicazione e quindi il PIC esce dalla funzione di gestione interrupt per continuare ciò che stava facendo prima dell'interruzione.

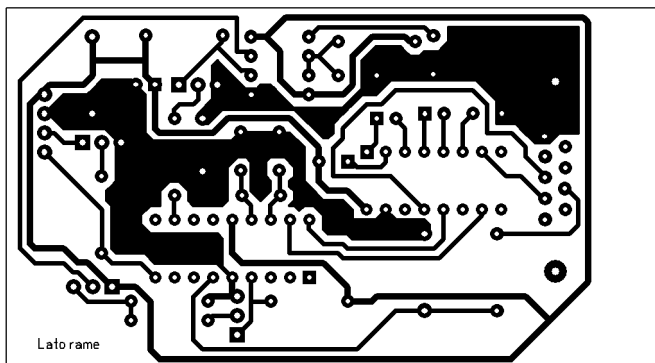



Figura 6 *Lato rame in scala 1:1*

Figura 7 *Lato componenti*

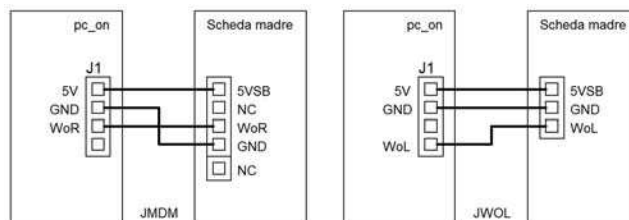



Figura 8 *Connessione circuito - Scheda madre*

MONTAGGIO DEL CIRCUITO

In figura 6 si ha il lato rame del circuito, come si può notare il montaggio non presenta particolari difficoltà, le piste sono sufficientemente larghe da resistere ai saldatori più rudi.

I condensatori elettrolitici intorno al Maxim (C2, C3, C4 e C5) possono anche avere altri valori fino ad un massimo di 10 µF ma devono comunque avere tutti lo stesso valore. I transistor non devono avere particolari requisiti dato che vengono utilizzati come interruttori quindi possono andare bene transistor NPN BF a patto che coincidano con la piedinatura del BC237.
Non si può assolutamente cambiare invece il

Elenco componenti	
Sigla	**Valore**
R1	4.7 KΩ 1/4 W
R2, R3, R5, R6, R8	10 KΩ 1/4 W
R4	22 KΩ 1/4 W
R7	100 KΩ 1/4 W
C1	100 μF 16V elettrolitico
C2, C3, C4, C5	1 μF 16V elettrolitico
C6,C7	22 pF ceramico
C8	22 nF poliestere
C9,C10	100 nF poliestere
U1	PIC16F84A
U2	NE555
U3	MAX232
X1	Quarzo 4MHz
Q1, Q2, Q3, Q4	BC237 npn
D1	Diodo Zener 5,1v
J1	Connettore linare a striscia F a 4 poli
J2	Connettore DB-9 Seriale F
Cavo	1 metro di cavo a tre poli
J3	Connettore a striscia M a 4 poli
Filo	Filo a tre poli
J4	Vedi testo

quarzo a 4MHz da cui dipende il conto alla rovescia del PIC ed una precisa comunicazione a 9600 baud, se infatti si mettesse un quarzo da 20MHz anziché decrementare ogni secondo il PIC decrementerebbe ogni 200ms ed inoltre non si riuscirebbe più a fa comunicare PC e PIC.
Bisogna fare anche attenzione nel fare il cavo che porta alla scheda madre, il connettore J1 infatti include sia il pin relativo al WoR che al WoL oltre che a 5VSB (alimentazione di 5V) e GND (riferimento di massa). Sul circuito il connettore femmina J1 ha quattro connessioni ma se ne utilizzeranno solo tre scelte a seconda che la scheda madre supporti il WoR od il WoL. Il J3 verrà saldato al filo a tre poli ed agganciato al J1 ed il filo si estenderà sino al J4 che dovrà incastrarsi con il connettore della scheda madre, per realizzare il filo completo bisognerà, quindi, prima sapere che tipo di connettore risiede sulla scheda che può essere in generale diverso da modello a modello. In molti casi J4 deve essere un connettore a striscia femmina da 3 o 5 poli a seconda che la scheda supporti rispettivamente WoL o WoM (come si vede in figura 8). In figura 8 si evidenziano le due possibili connessioni circuito (pc_on) e scheda madre.

La scelta del PIC16F84A ha lo svantaggio di non offrire tutta una serie di caratteristiche tecniche che avrebbero portato ad una semplificazione del firmware, ma ha il vantaggio di essere facilmente reperibile, ben conosciuto ed è possibile utilizzare praticamente tutti i programmatori per microcontrollori Microchip. Inoltre lo sviluppo di un particolare settore dell'elettronica, qualche anno fa, ha portato in molte case diversi tipi di programmatori molto semplici e specifici per questo microcontrollore (ad esempio il Ludipippo o affini) e nel caso proprio non se ne sia in possesso la realizzazione di questi è molto semplice e la documentazione tra le più complete.

UTILIZZO DEL SOFTWARE

Il software Kalendarjo così come il firmware del PIC sono scaricabili dal sito di Fare Elettronica.
Per ciò che riguarda il software Kalendarjo.exe bisogna avviare l'eseguibile e procedere seguendo le istruzioni di installazione, completato questo si potrà notare che il software rimane attivo e visibile nella tray-bar (in basso a destra a lato dell'orologio). Si apra la finestra

Figura 9 *Finestra principale di Kalendarjo.exe*

principale clikkando sull'icona in arancione che indica la data odierna, apparirà la finestra di figura 9.
Premendo l'icona indicata dalla freccia rossa con "Programmazione spegnimento PC" si apre una finestra su cui dovranno essere inseriti i dati relativi ad uno spegnimento programmato del PC. Mentre premendo l'icona indicata con "Programma accensione PC" si aprirà la finestra di figura 10, simile alla precedente, in cui si potranno gestire le informazioni necessarie all'avvio automatico del computer.
Appare come indicato dalla freccia rossa il record che contiene tutte le informazioni che sono in relazione con la programmazione: data di accensione, programma da avviare, titolo che descrive brevemente il record e, anche se nascosto, l'informazione che indica una connessione forzata ad internet.
Il menù indicato con la freccia azzurra ha cinque tasti che svolgono le seguenti funzioni:

- Programma: apre l'immagine di figura 11, in cui bisogna inserire i dati che appariranno in seguito nel record.
- Modifica: consente di modificare il record.
- Cancella: effettua una cancellazione del record selezionato ed in seguito cancella anche i dati residenti sul PIC.
- Stato: consente di osservare a che punto si trova il conteggio del PIC.
- Si limita a cancellare le informazioni sul PIC, ma attenzione, nel caso siano presenti record nella griglia, automaticamente, Kalendarjo programma il PIC con quello più recente.

Figura 10 Finestra accensione PC di Kalendarjo.exe

Figura 11 Finestra programmazione PIC

In figura 11 si osservano tutti i campi da modificare e che consentono una programmazione completa.
Non sono le uniche funzioni che ha questo software, comprende, infatti, una sezione dedicata alla memorizzazione di appuntamenti e di dati in genere che non verrà approfondita ma lasciata alla curiosità di chi ne farà uso.

CONCLUSIONI

La parte elettronica è resa il più semplice possibile e costituita da una parte centrale importante che è il microcontrollore responsabile di tutto il funzionamento del circuito, questa centralizzazione ha anche il vantaggio di rendere possibili e più semplici da realizzare eventuali modifiche di sostanza al funzionamento del circuito. Ad esempio si può pensare di estendere il conteggio a 4 Byte portando la capacità di controllo da circa 193 giorni a circa 136 anni.
Il software ed il Firmware del PIC sono disponibili sul sito di Fare Elettronica.

Figura 12 Come si presenta il circuito

fare
elettronica

L'amplificatore operazionale dalla A alla Z

Come già per l'op-amp in configurazione non invertente, si espongono qui le corrispondenti relazioni per la configurazione invertente

6.1 L'OPERAZIONALE INVERTENTE. IL GUADAGNO

L'applicazione della reazione di tensione parallelo si ha nella configurazione invertente dell'amplificatore operazionale nella quale il segnale pilota è applicato all'input invertente, mentre l'input non invertente è collegato a massa come mostra la figura 6.1. La rete di reazione è costituita dalla resistenza R_f che connette l'uscita all'ingresso invertente, e dalla resistenza R_i il cui valore stabilisce la resistenza di ingresso dello stadio.
Per giungere all'espressione del guadagno si inizia, sempre con riferimento alla figura 6.1, col considerare nulla la corrente entrante nell'input invertente dell'amplificatore operazionale.
Questa condizione implica che la corrente I_r che fluisce nella resistenza R_i sia la stessa che passa nella resistenza R_f. Pertanto, dall'eguaglianza fra le correnti si ha:

$$(V_i - V_b) / R_i = (V_b - V_o) / R_f \qquad [6.1]$$

D'altra parte per il guadagno ad anello aperto vale la relazione nota:

$$A_{OL} = V_o / (V_a - V_b)$$

Ma per $V_a = 0$ (ingresso non invertente a massa) l'espressione precedente diventa:

$$A_{OL} = - V_o / V_b$$

Da questa si ricava:

$$V_b = - V_o / A_{OL}$$

Sostituendo quindi nella [6.1] si ha:

$$[V_i + (V_o / A_{OL})] / R_i = [(-V_o/A_{OL}) - V_o) / R_f] \qquad [6.2]$$

Ricavando da questa il rapporto $A_{CL} = V_o / V_i$ si ottiene:

$$A_{CL} = \frac{V_o}{V_i} = -\frac{R_f A_{OL}}{R_i + A_{OL} R_i + R_f} = -\frac{R_f}{\frac{R_i}{A_{OL}} + R_i + \frac{R_f}{A_{OL}}} \qquad [6.3]$$

Per $A_{OL} \to \infty$, si ha infine:

$$A_{CL} = \frac{V_o}{V_i} = -\frac{R_f}{R_i} \qquad [6.4]$$

Il guadagno dell'amplificatore operazionale in configurazione invertente è quindi eguale, in modulo, al rapporto fra la resistenza R_f e la resistenza R_i. Pertanto, come nel caso della configurazione non invertente, dipende soltanto dal valore attribuito a queste due resistenze ed è pressoché indipendente dall'operazionale utilizzato.
I segno (–) che compare nell'espressione [6.4] indica che il segnale di uscita è in opposizione di fase rispetto al segnale di ingresso. All'espressione [6.4] si può pervenire più rapidamente, come già si è illustrato per l'amplificatore non invertente, tramite il concetto di massa virtuale, ossia attraverso l'equipotenzialità degli ingressi dell'amplificatore operazionale. Con riferimento alla figura 6.1, essendo l'ingresso non invertente a massa, ossia a potenziale zero, allo stesso potenziale zero dev'essere l'ingresso invertente. L'espressione [6.1] diviene allora:

$$(V_i - 0) / R_i = (0 - V_o) / R_f$$

Da questa si ha immediatamente:

$$A_{CL} = V_o / V_i = - R_f / R_i$$

L'op-amp invertente

di Nico Grilloni
n.grilloni@farelettronica.com

Il rapporto V_o/V_i della [6.3] può anche porsi nella forma:

$$A_{CL} = \frac{V_o}{V_i} = -\frac{R_f}{R_i} \cdot \frac{A_{OL}}{1 + \frac{R_f}{R_i} + A_{OL}}$$

che, tenendo conto dell'attenuazione del blocco di reazione $\beta = R_i/(R_i + R_f)$, da cui si ricava il rapporto $R_f/R_i = (1 - \beta)/\beta$, diventa:

$$A_{CL} = \frac{V_o}{V_i} = -\frac{R_f}{R_i} \cdot \frac{1}{1 + \frac{1}{A_{OL}\beta}} = -\frac{R_f}{R_i} \cdot E_r \qquad [6.5]$$

Da questa espressione, che per $A_{OL} \rightarrow \infty$ coincide con la [6.4], si vede che il guadagno dell'amplificatore operazionale invertente, al pari dell'amplificatore operazionale non invertente, è il prodotto del guadagno ideale (R_f/R_i) per un termine E_r di errore. Se il guadagno di anello ($A_{OL} \cdot \beta$) tende a un numero molto grande, allora E_r tende a uno e il guadagno reale approssima il guadagno ideale. In termini pratici è necessario avere alti valori del prodotto ($A_{OL} \cdot \beta$) e ciò implica l'utilizzazione di op-amp ad alto A_{OL}. Infatti, nel caso di guadagni ragguardevoli, ossia per valori elevati del rapporto (R_f/R_i), β è sempre un numero piccolo come meglio si vede ponendo β nella forma:

$$\beta = \frac{1}{1 + \frac{R_f}{R_i}} \qquad [6.6]$$

Pertanto, nei casi in cui si richiedano elevati valori di A_{CL}, l'unica via percorribile è quella che fa ricorso ad operazionali ad elevato A_{OL}.

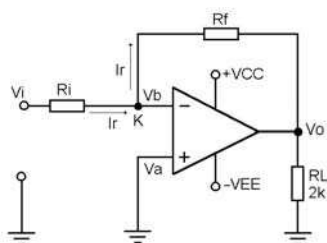



Figura 6.1 Amplificatore operazionale in configurazione invertente

ESEMPIO 6.1

Si dimensioni, utilizzando l'operazionale LM833, lo stadio amplificatore in configurazione invertente della figura 6.1 per un guadagno $A_{CL} = 10$.

Soluzione

Si potrà, per esempio, attribuire alla resistenza R_f il valore di 22 kΩ. Dalla [6.4] si ha quindi per R_i:

$$R_i = R_f / A_{CL}$$

Per la resistenza R_i si avrà allora:

$$R_i = 22000 / 10 = 2{,}2\ \text{k}\Omega$$

La figura 6.2 riporta l'andamento delle forme d'onda relative allo stadio qui dimensionato. Si noti lo sfasamento di 180° fra ingresso e uscita e l'amplificazione pari a 10 apportata dallo stadio al segnale V_i di ingresso. A 2 V da picco a picco corrispondono infatti, in uscita 20 V da picco a picco. L'ampiezza della V_o è evidenziata dalla posizione reciproca dei marker orizzontali. Si legge infatti *(c – d)* = 19,98 V. La distanza *(a – b)* fra i marker verticali indica la frequenza (1 kHz).
La figura 6.3 riporta quindi la curva di risposta in frequenza. A –3 dB, indicati dalla distanza *(c – d)*, il marker verticale *a* indica la frequenza di taglio f_H = =1,259 MHz. L'ordinata *Yd* = 20, ossia la distanza fra il marker *d* e il livello 0 dB del segnale di ingresso indica il guadagno. A 20 dB corrisponde, infatti, un valore di A_{CL} pari a 10. Si può infine calcolare il valore del fattore di errore E_r presente nell'espressione [6.5]. Essendo, per l'LM833, A_{OL} = 110 dB (pari a $10^{5,5}$) e β =0,09, il termine:

$$E_r = \frac{1}{1 + \frac{1}{A_{OL}\beta}}$$

assume il valore:

$$E_r = \frac{1}{1 + \frac{1}{10^{5,5} \times 0{,}09}} = 0{,}99$$

Anche in questo caso E_r tende a uno e si può pertanto ritenere trascurabile. Il guadagno reale si può pertanto considerare coincidente col guadagno ideale.

6.1.1 La banda passante nell'op-amp reazionato invertente

Per la determinazione della frequenza di taglio superiore f_H che rappresenta l'estremo superiore della banda passante a –3 dB, si procede come già per lo stadio non invertente. Pertanto, eguagliando il prodotto *GBW*, guadagno per banda passante, dell'amplificatore ad anello aperto e dell'amplificatore ad anello chiuso, e ricavando quindi f_H, si ha l'espressione:

$$f_H = f_T \cdot \frac{A_{OL}}{A_{CL}} \qquad [6.7]$$

dove f_t è la frequenza di taglio caratteristica dell'operazionale utilizzato ad anello aperto e f_H è la frequenza di taglio superiore dello stadio amplificatore. Sostituendo nella [6.5] al rapporto (R_f/R_i) il termine equivalente $(1 - \beta)/\beta$, si ha l'espressione:

Figura 6.2 *Forme d'onda in ingresso e in uscita nell'amplificatore invertente della figura 3.1. La tensione Vi di ingresso ha un'ampiezza di 2 V da picco a picco. La tensione V_o di uscita ha un'ampiezza di circa 20 V dapicco a picco. Uscita e ingresso sono in opposizione di fase*

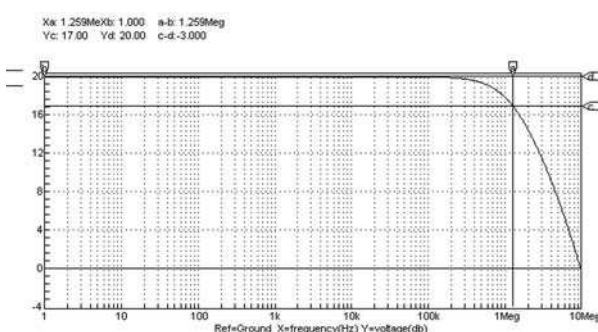



Figura 6.3 *Curva di risposta in frequenza dello stadio amplificatore di cui alla figura 3.1. La banda passante si estende da 0 Hz fino alla frequenza di taglio superiore f_H = 1,279 MHz. L'ordinata Yd = 20 indica il guadagno in dB. A 20 dB corrisponde infatti, un valore di A_{CL} = 10*

$$A_{CL} = \frac{1-\beta}{\beta} \cdot \frac{1}{1+\frac{1}{A_{OL}\,\beta}} = \frac{(1-\beta)\,A_{OL}}{1+A_{OL}\,\beta} \qquad [6.8]$$

Sostituendo questa nella [6.7] si ricava:

$$f_H = f_t \cdot \frac{1+A_{OL}\,\beta}{1-\beta} \qquad [6.9]$$

Essendo $\beta = R_i/(R_i + R_f)$ si ha:

$$\frac{1}{\beta} = \frac{R_i + R_f}{R_i} = 1 + \frac{R_f}{R_i} = 1 + A_{CL}$$

Da cui:

$$\beta = \frac{1}{1+A_{CL}} \qquad [6.10]$$

Sostituendo la [6.10] nella [6.9] si ottiene l'espressione:

$$f_H = f_i \cdot \left[\frac{1+A_{OL}}{A_{CL}} + 1\right] \qquad [6.11]$$

che esprime la frequenza di taglio superiore, ossia, in questo caso, la banda passante *B*, in funzione della frequenza di taglio f_t caratteristica dell'operazionale, del guadagno A_{OL} ad anello aperto e del guadagno A_{CL} ad anello chiuso. Le espressioni [6.7], [6.9] e [6.11], pertanto, sono utili per calcolare la frequenza di taglio superiore in uno stadio amplificatore con op-amp in configurazione invertente. Si prenda in considerazione, per esempio, l'amplificatore su dimensionato per un guadagno A_{CL} = 10 tenendo conto che in simulazione al computer l'LM833 ha fornito le seguenti caratteristiche: A_{OL} = 113 dB (e non 110 dB come nei dati della Casa), GBW = 10 MHz e f_t = 28,6 Hz. Tramite la [6.7] si ricava:

$$f_H = 28{,}6 \times (10^{5{,}65}/10) = 1{,}277 \text{ MHz}$$

Tramite la [6.9], tenendo conto che β = 0,09 e quindi $(1 - \beta)$ = 0,91, si ha:

$$f_H = 28{,}6 \times \frac{1+(10^{5{,}65} \times 0{,}09)}{0{,}91} = 1{,}26 \text{ MHz}$$

Tramite la [6.11]:

$$f_H = 28{,}6 \times \left[\frac{1+10^{5{,}65}}{10} + 1\right] = 1{,}27 \text{ MHz}$$

Si noti come il valore della frequenza f_H desunto dal calcolo sia pressoché coincidente col valore ottenuto in simulazione.

6.1.2 La resistenza di ingresso dell'op-amp invertente reazionato

Questa resistenza assume il valore attribuito alla resistenza R_i. Infatti, applicando il concetto di massa virtuale, il terminale dell'ingresso invertente – punto *K* nella figura 6.1 – è all'incirca allo stesso potenziale zero dell'ingresso non invertente che è connesso a massa. La resistenza di ingresso dello stadio amplificatore coincide quindi con il valore dato alla resistenza R_i.
Si ha pertanto:

$$R_{iCL} = R_i << R_{in} \qquad [6.12]$$

Nel dimensionamento sarebbe quindi opportuno calcolare prima la R_i in funzione della resistenza di ingresso voluta e quindi la resistenza R_f in funzione del guadagno. Dall'espressione [6.12] si deduce che, contrariamente all'amplificatore operazionale in configurazione non invertente, *nella configurazione invertente si ha una resistenza di ingresso minore della resistenza di ingresso intrinseca R_{in} dell'operazionale utilizzato.* Pertanto, a meno di non attribuire alla R_i un valore elevato, che comporterebbe, in funzione del guadagno, un valore ancora più elevato della R_f,sarà sempre $R_i<<R_{in}$.
Si tenga presente che per il valore da attribuire alla resistenza R_f vale quanto già detto per la configurazione non invertente.

6.1.3 La resistenza di uscita dell'op-amp invertente reazionato

Per l'espressione analitica della resistenza di uscita R_{oCL} dell'amplificatore operazionale in configurazione invertente valgono le stesse considerazioni esplicitate per la configurazione non invertente. Si giunge pertanto all'espressione:

$$R_{oCL} = R_o / (1 + A_{OL}\ \beta)$$

che esprime il concetto secondo il quale la resistenza di uscita dell'amplificatore reazionato è sempre molto minore della resistenza intrinseca di uscita dell'operazionale. Si ha quindi:

$$R_{oCL} << R_o$$

Per quanto riguarda la resistenza di uscita non c'è quindi alcuna differenza di comportamento fra la configurazione invertente e la non invertente dell'amplificatore operazionale.

6.1.4 Azzeramento della tensione di offset

Nel caso in cui sia necessario ricorrere alla compensazione esterna per annullare la tensione di offset, come può accadere negli amplificatori in continua, si potrà ricorrere ad una delle due configurazioni esposte nella figura 6.4.
Nella prima – figura 6.4 *a* – la rete di compensazione è posta sull'ingresso invertente, ossia sullo stesso ingresso del segnale; nella seconda – figura 6.4 *b* – la rete di compensazione è posta sull'ingresso non invertente. In entrambi i casi il guadagno ad anello chiuso è comunque fornito dall'espressione (R_f/R_i) caratteristica della configurazione invertente. Si fa, infine, notare che normalmente le Case costruttrici forniscono tutti i dati utili alla realizzazione delle reti per l'annullamento dell'offset.

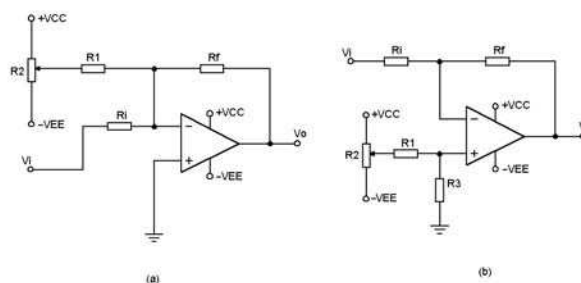



Figura 6.4 *Amplificatore con operazionale in configurazione invertente con rete di compensazione per la tensione di offset*

6.1.5 Diminuzione della corrente di offset

Valgono le stesse considerazioni fatte per l'amplificatore operazionale in configurazione non invertente. Da un punto di vista circuitale la resistenza R_p, di valore eguale al parallelo $R_i//R_f$, nella configurazione invertente va posta fra l'ingresso non invertente e massa così come mostra la figura 6.5.

ESEMPIO 6.2

Si dimensioni un amplificatore in configurazione invertente – figura 6.1 – avente un guadagno pari a 20 e se ne calcoli la banda passante.
Si faccia uso dell'op-amp LF353 alimentato a ± 15 V.

Soluzione
Si potrà porre R_f = 47 kΩ. Per A_{CL} = 20, dalla [3.4] si ha per R_i:

$$R_i = 47000/20 = 2350\ \Omega \quad \Rightarrow$$
$$(R_i = 2 \text{ resistenze da } 4{,}7\ k\Omega \text{ in parallelo})$$

Usando l'LF353, il cui stadio differenziale di ingresso è a Fet, si può fare a meno della resistenza R_p fra il pin non invertente e massa. Se però si utilizzasse un op-amp con stadio differenziale di ingresso a BJT sarebbe necessario porre fra l'input non invertente e massa una resistenza $R_p = R_i//R_f$ al fine di limitare la corrente di offset e quindi la conseguente tensione di offset. Per il calcolo della banda passante, ovvero della frequenza di taglio f_H si può fare ricorso all'espressione [6.9] tenendo conto che il guadagno ad anello aperto A_{OL} dell'LF353 è di 100 dB (10^5) e che la sua frequenza di taglio intrinseca f_t vale 32,45 Hz.
Essendo β = 2350/49350 = 0,0476 si ha:

$$f_H = 32{,}45 \cdot \frac{1 + (10^5 \times 0{,}0476)}{1 - 0{,}0476} = 162{,}2\ kHz$$

La figura 6.6 espone la curva di risposta in frequenza dell'amplificatore invertente qui dimensionato. La frequenza di taglio superiore ha valore pressoché eguale a quello su calcolato. Infatti, a –3 dB il marker verticale *a* indica la frequenza f_H = 163,5 kHz. L'ordinata *Yd* = 26,01 dB indica il guadagno. A 26,01 dB corrisponde un guadagno puramente numerico pari a 19,95.

Figura 6.5 *Nella configurazione invertente, per diminuire la tensione di offset dovuta alla corrente di offset, la resistenza R_p va posta fra l'input non invertente e massa.*

6.2 Incremento della resistenza di ingresso

Il dimensionamento dell'amplificatore di cui all'esempio precedente, si è iniziato attribuendo alla resistenza R_f un valore non eccessivo (47 kΩ). Quindi, in funzione del guadagno, si è ricavato il valore della resistenza R_i. Questo procedimento è in linea di massima valido e da tenere quindi in considerazione quando la resistenza della sorgente può ritenersi trascurabile. In questo caso, infatti, per il trasferimento del segnale dalla sorgente all'input invertente dell'op-amp, può essere accettabile anche un basso valore della resistenza di ingresso dello stadio che, come si è visto, coincide col valore della R_i (2,35 kΩ nell'esempio).
Nel caso, invece, in cui la sorgente di segnale dovesse avere una resistenza propria non trascurabile, il problema, almeno in prima approssimazione, potrebbe risolversi iniziando il dimensionamento dalla R_i in funzione della resistenza interna R_s della sorgente di segnale.
Nell'ipotesi, per esempio, che sia R_s = 10 kΩ, la resistenza di ingresso dello stadio amplificatore dovrà essere almeno dieci volte maggiore. Il che impone che sia R_i = 100 kΩ. Ma per un guadagno pari a 20, alla resistenza R_f dovrebbe essere attribuito il valore di 2 MΩ e per un guadagno pari a 100 si dovrebbe porre in circuito una R_f = 10 MΩ. Questo procedimento è del tutto in contrasto con quanto si è detto a proposito del valore non eccessivo da doversi attribuire alla resistenza R_f.
La soluzione migliore non può quindi essere quella che porta a un incremento della R_i. Molto meglio, pertanto, ricorrere alla configurazione riportata nella figura 6.7 nella quale, a monte dell'amplificatore, si è posto un inseguitore di tensione.
Come è noto, la resistenza di ingresso dell'inseguitore è sempre elevatissima, e pertanto sempre in grado di superare, e di molto, il valore della resistenza della sorgente. In pratica, l'interposizione dell'inseguitore di tensione svincola la resistenza R_s della sorgente dalla resistenza R_i di ingresso dello stadio amplificatore.
Così facendo alla R_i (e quindi alla R_f) si potrà attribuire un valore anche estremamente modesto.

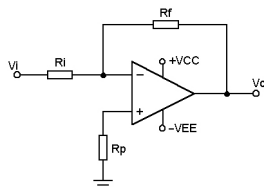
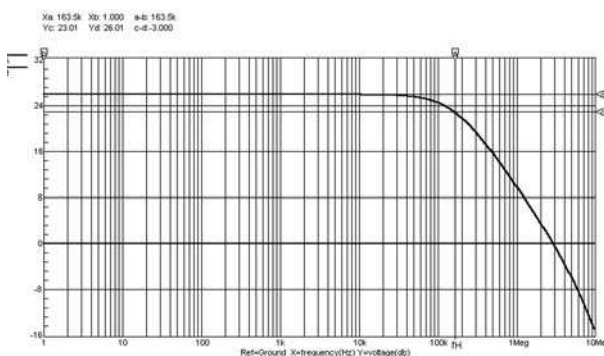



Figura 6.6 *Curva di risposta dell'amplificatore di cui all'esempio 3.2 con guadagno A_{CL} = 20 e frequenza di taglio superiore f_H = 163,5 kHz.*

6.3 VERIFICA DELLA TENSIONE DI OFFSET DI USCITA TRAMITE SPICE

Come già per la configurazione non invertente, anche per l'invertente, attraverso la simulazione con SPICE si può prendere atto della diminuzione della tensione offset di uscita determinata dalla presenza della resistenza R_p fra l'ingresso non invertente e massa.

Con riferimento all'amplificatore dimensionato nell'esempio 6.1, la simulazione con SPICE fornisce, infatti, le tensioni evidenti nella figura 6.8. In assenza della R_p – figura 6.8 a – la tensione di offset di uscita è pari a –14 mV, mentre in presenza della resistenza R_p la medesima tensione si porta a –136 μV. In questo caso la presenza della resistenza $R_p = R_f // R_i$ riduce quindi la tensione di offset di uscita di oltre cento volte.

PER IL LETTORE

I diagrammi qui riportati sono stati ricavati con il software di simulazione CircuitMaker della Microcode Engineering Inc. – Utah – USA, che utilizza SPICE (*Simulation Program with Integrated Circuit Emphasis*) realizzato dalla Microcode Engineering Inc. – Utah – USA. Sito Internet: microcode.com.

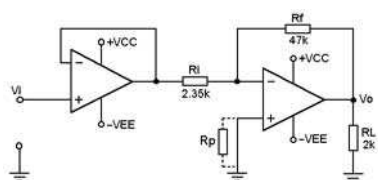



Figura 6.7 Stadio amplificatore non invertente preceduto da un inseguitore di tensione per garantire una resistenza di ingresso indipendente dalla R_i

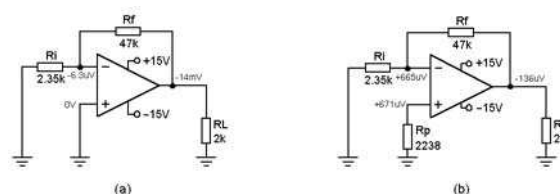



Figura 6.8 L'inserimento della resistenza R_p riduce notevolmente la tensione di offset di uscita. In a, ossia in assenza della R_p, si ha una tensione di offset di –14 mV, mentre in b, ossia in presenza della Rp, si ha una tensione di offset di 136 μV

gli appuntamenti

10-11 Settembre 2005

32ª TELERADIO — Piacenza

Teleradio è fra le più prestigiose mostre mercato italiane specializzate nel campo delle telecomunicazioni. I settori trattati sono: materiale radiantistico per radio-amatori e C.B., apparecchiature per telecomunicazioni, surplus - telefonia - elettronica e computers, antenne e parabole satellitari per ricezione TV, antenne e parabole per radio-amatori.

Luogo: Piacenza Expo (Quartiere Fieristico di Piacenza) Piacenza
Orari: 9:00 - 18:00
Organizzatore: Teleradio (www.teleradio.piacenzaexpo.it Tel. 0523 602711)
Ingresso: € 6,00

17-18 Settembre 2005

MOSTRA MERCATO DELL'ELETTRONICA APPLICATA RADIOAMATORE C.B. — Macerata

Mostra mercato specializzata nei seguenti settori: elettronica, radioamatore C.B., Hi Fi, computer.

Luogo: Centro fiere Villa Potenza - Macerata
Orari: 8:30 - 19:30
Organizzatore: C.B. Club Maceratese (www.cbclubmaceratese.com - Tel 0733.493067)
Ingresso: n.p.

17-18 Settembre 2005

EXPO ELETTRONICA & COLLEZIOSA — Rimini

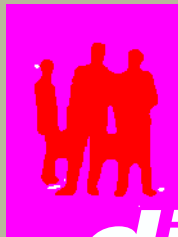
Expo Elettronica è la mostra mercato dedicata all'elettronica e punto d'incontro fra "antiquariato tecnologico" e applicazioni "futuribili": una miriade di oggetti e applicazioni ormai indispensabili come computer, software, periferiche, home theater, telefonia fissa e mobile, accessori, ricambi, curiosità elettroniche e digitali. Parallelamente si svolge Colleziosa, dove si trova di tutto un po': vecchie radio, macchine fotografiche, dischi d'epoca e cd rari, fumetti, numismatica, filatelia, modellismo, giocattoli, soldatini, sorpresine, ed altro ancora, rigorosamente usato e da collezione.

Luogo: Palacongressi della Riviera di Rimini - Rimini
Orari: 9:00 - 18:00
Organizzatore: Blu Nautilus (www.blunautilus.it)
Ingresso: € 6,50 (ridotto € 5,00)

17-18 Settembre 2005

FIERA DEL RADIOAMATORE — Monterotondo (ROMA)

Fiera focalizzata sui seguenti settori: ricetrasmittenti, apparati VHF, apparati C.B., componentistica, valvole, antenne, paraboliche, utensili per hobby, computer, strumentazione, schemi elettronici, editoria, programmi aziendali, video giochi, radio d'epoca.

Luogo: Palazzetto dello Sport - Monterotondo (Roma)
Orari: 9:00 - 19:00
Organizzatore: Gedit (gedit2004@libero.it - Tel. 348.3307007)
Ingresso: n.d.

24-25 Settembre 2005

FIERA DELL'ELETTRONICA E DEL RADIOAMATORE — Gonzaga (MN)

I visitatori della Fiera dell'Elettronica e del Radioamatore, potranno trovare una vastissima gamma di prodotti tra i quali spiccano: materiale elettronico, computer, software, materiale radioimpiantistico, surplus, componentistica, telematica, decoder, parabole e telefonia con la partecipazione dell'editoria specializzata. La rassegna fornirà anche una panoramica didattico-educativa con la presenza di numerosi editori di riviste e testi specializzati.

LUOGO: Parco Fiera Millenaria Gonzaga (MN)
ORARI: 8:30 - 18:00
ORGANIZZATORE: Fiera Millenaria di Gonzaga S.r.l. (www.fieramillenaria.it - Tel 0376.58098)
INGRESSO: € 7,00 (ridotto € 5,00)

29 Sett. - 02 Ott. 2005

2a ELETTRONIKA — Ancona

Elettronika 2005 sarà caratterizzata dalla partecipazione di aziende che rappresentano le eccellenze nell'ambito Research & Development, con l'obiettivo di favorire e consolidare posizioni di leadership tecnologia, know how e produttività nel nostro sistema industriale. Punto focale della manifestazione sarà evidenziare le iniziative di innovazione e sviluppo in ambito di prodotti, tecnologie, processi e materiali, nei settori chiave dell'elettronica quali: ICT, Componentistica, Applicazioni Scientifiche e Medicali, Automazione Industriale, Test e Produzione, Outsourcing e Subcontracting.

LUOGO: Quartiere fieristico di Ancona - Ancona
ORARI: 9:30 - 18:30
ORGANIZZATORE: ERF (www.elettronika.net - Tel 071.58971)
INGRESSO: Gratuito, previa registrazione

29 Sett. - 01 Ott. 2005

12a EDIZIONE SAT EXPO — Vicenza

SAT Expo è la vetrina di riferimento per le comunicazioni via satellite e per la Tv digitale terrestre. È una manifestazione che presta grande attenzione e spazio anche alle soluzioni wireless, all'automazione domestica, al cablaggio strutturato, quindi a tutte le più avanzate e innovative tecnologie di comunicazione oggi sul mercato o in fase di test.

LUOGO: Fiera di Vicenza - Vicenza
ORARI: 9:00 - 18:00
ORGANIZZATORE: Promospace (www.satexpo.it)
INGRESSO: Riservato agli operatori, previa registrazione

01-02 Ottobre 2005

RADIANT AND SILICON — Segrate (MI)

La Mostra comprende tra le altre le seguenti voci merceologiche: apparecchi e componenti per le telecomunicazioni, ricetrasmissioni, elettronica, informatica, videogiochi, surplus, editoria specializzata, radio d'epoca.
Attualmente RADIANT, che si sviluppa su una superficie espositiva di 10.000 mq., conta più di 160 espositori ed oltre 12.000 visitatori.

LUOGO: Parco Esposizioni Novegro - Segrate (MI)
ORARI: 9:00 - 18:00
ORGANIZZATORE: Comis - Parco Esposizioni Novegro (www.parcoesposizioninovegro.it)
INGRESSO: € 8,00

Mikrobasic per PICmicro

Studiamo insieme il PWM, una tecnica molto utile dalle applicazioni più svariate. Utilizzando il Mikrobasic scopriremo come si può gestire e personalizzare questa tecnica al massimo livello, permettendci di accrescere il valore professionale delle nostre realizzazioni.

Il nostro microcontrollore è capace di generare solamente due livelli di tensione in uscita: 0 Volt (livello logico basso) e 5 Volt (livello logico alto). Non esistono mezze misure, nessun compromesso e nessuna altra tensione compresa tra i due limiti: solo questi due valori. Alcune volte però è necessario generare una tensione minore, per poter pilotare, con meno corrente, un determinato carico. Se colleghiamo una lampadina all'uscita del PIC, essa (normalmente) potrà assumere la condizione di spenta (se all'uscita è applicato uno 0 logico) o di accesa (se all'uscita è applicato un 1 logico). Dovendo però illuminare la lampada con minore intensità, è necessario far ricorso alla tecnica del PWM. Solo in questa maniera potremo "dosare" finemente la tensione prodotta, in modo che la stessa lampada può assumere diversi gradi di illuminazione.

IL PWM

PWM è l'acronimo dei termini inglesi *Pulse With Modulation*, ossia impulsi modulati. Si tratta in sostanza di una tecnica geniale, che consente di variare la potenza applicata ad un carico, non modificando la tensione erogata sull'utilizzatore bensì "variando" la percentuale della larghezza degli impulsi applicati al carico stesso. Se quindi si desidera che una lampadina si illumini con diverse intensità, non bisogna cambiare la tensione su di essa ma occorre semplicemente variare (con molta velocità) il tempo durante il quale la tensione massima viene applicata al carico. In sostanza esso viene sottoposto ad un "bombardamento" di impulsi positivi, ossia un **treno di impulsi**, opportunamente calibrato.
È indubbiamente un metodo più complicato, ma permette di ridurre al minimo la dissipazione di eventuali transistor di potenza utilizzati, in quanto si comportano da semplici interruttori e non da regolatori di corrente.
Guardiamo per un attimo lo schema ideale di figura 1. Esso è formato da un generatore di treno di impulsi ad onde quadre P (V1) e da un carico rappresentato da una lampada ad incandescenza (L1). Il generatore d'impulsi fornisce alla lampada solamente dei segnali ON-OFF cadenzati, con determinate temporizzazioni. Con tale tecnica è possibile dosare con continuità la potenza fornita a lampade, motori, led e tanti altri carichi utilizzatori. Alimentando i motori in corrente continua è possibile rallentarne il numero dei giri in modo drastico mantenendo inalterata la coppia specifica, operazione sicuramente impossibile variando solamente la tensione tipica di alimentazione. Esaminiamo alcuni casi.

Duty Cycle al 50%

Il generatore fornisce, nell'esempio, alcuni impulsi ON-OFF con periodo equivalenti, vale a dire 1 millisecondo a livello basso ed 1 millisecondo a livello alto. La lampadina pertanto

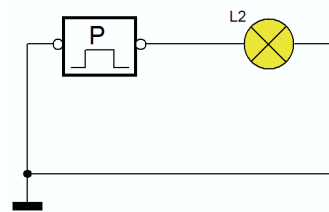



Figura 1 Schema teorico di generatore di impulsi e carico

PWM con il PIC

di Giovanni Di Maria
g.dimaria@farelettronica.com

riceve un treno di impulsi bilanciato, come visibile dal grafico in figura 2. Si illumina quindi con una intensità *dimezzata* rispetto al suo massimo teorico, proprio per il fatto che riceve alternativamente segnali di acceso e di spento. Il ciclo di lavoro dell'onda quadra generata, ossia il suo *duty cycle*, è pertanto del 50%. Il carico, in questo caso, è come se fosse alimentato da una tensione divisa per due, rispetto alla sorgente di alimentazione, con conseguente diminuzione della luminosità o della sua resa in genere.

Duty Cycle al 10%

Se il generatore fornisce un impulso positivo (attivo) della durata di 1 millisecondo ed i rimanenti 9 millisecondi restano inattivi (0 Volt) il duty cycle è pari al 10%. In altre parole, nell'ambito dell'intero periodo della durata di 10 millisecondi, esiste solamente un impulso "produttivo" della durata di 1 ms. (equivalente quindi al 10%). Gli altri 9 ms. costituiscono un periodo di inattività, di tempo morto, nel quale il carico non riceve corrente e quindi non è alimentato. La lampada si illumina in questo caso molto debolmente, al limite minimo della sua incandescenza. In figura 3 è riportato il grafico degli impulsi generati (impulsi stretti).

Duty Cycle al 90%

Se la larghezza attiva dell'impulso è di 9 ms. su un periodo totale di 10 ms. avremo in questo caso un duty cycle del 90%. Il carico resta alimentato per la maggior parte del tempo e fornisce una resa quasi ideale. La lampada si accenderà quasi al massimo. In figura 4 riportiamo il grafico dei relativi impulsi (impulsi larghi).

Tabella di conversione da percentuale del duty cycle a Volt in uscita da una tensione iniziale di picco di 5 Volt

Duty Cicle (in percentuale %)	Valore efficace ottenuto (in Volt)
0	0,0
10	0,5
20	1,0
30	1,5
40	2,0
50	2,5
60	3,0
70	3,5
80	4,0
90	4,5
100	5,0

Figura 2 Duty Cycle al 50%

Figura 3 Duty Cycle al 10%

Figura 4 Duty Cycle al 90%

DISCO ACROMATICO

Per comprendere meglio il meccanismo del PWM possiamo eseguire l'esperimento del *disco acromatico*. Un disco acromatico è un cerchio rotante, suddiviso in tanti settori circolari alternati, di colore bianco e nero.
Il colore bianco indica (nel nostro caso) lo stato logico alto e quindi il valore analogico di 5 Volt. Il colore nero indica lo stato logico basso e quindi il valore analogico di 0 Volt. Facendo ruotare molto velocemente il disco, esso assume una graduale colorazione grigia, benché tale colore non sia presente.
La densità del grigio è proporzionale alle quantità dei settori bianchi e neri utilizzati. Un grigio scuro indica maggior presenza di aree nere, un grigio chiaro invece indica maggior presenza di aree bianche. Nelle figure 5, 6 e 7 possiamo studiarne il comportamento.

Figura 5 *Disco acromatico al 50%*

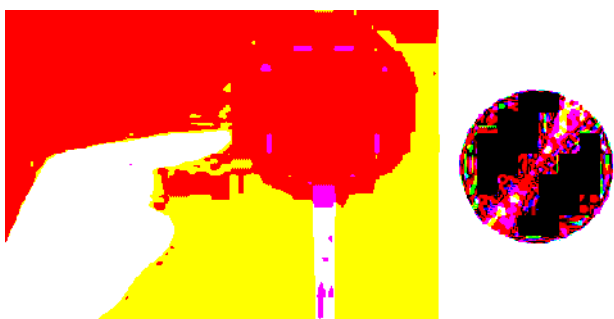



Figura 6 *Disco acromatico al 10%*

Figura 7 *Disco acromatico al 90%*

PRIMA ESPERIENZA

Sappiamo bene che le uscite del PIC possono generare solamente una tensione di 5 Volt per il livello logico alto ed una tensione di 0 Volt (tensione a massa) per il livello logico basso. Vedremo, con questo esperimento, come sia invece possibile ottenere, ad esempio, 3 Volt dalla porta RB0, con la scrittura di un opportuno programma. Realizziamo lo schema elettrico, peraltro molto semplice, della figura 8.

Come si ottengono i 3 Volt

Abilitando perennemente il pin RB0 otterremmo da questo il potenziale di 5 Volt (duty cycle 100%). Ma siccome è nostro scopo quello di ottenere solamente 3 Volt, è ovvio che dovremo sottoporre tale pin ad una cadenza alternata ed estremamente veloce di impulsi. Per conoscere la percentuale del duty cycle da impostare, basta eseguire la proporzione *%ON = (100 x Volt) : 5* dove Volt corrisponde alla tensione efficace che si vuole ottenere in uscita. Sostituendo i valori con quelli dell'esercizio proposto, il risultato finale ci fornisce la percentuale di duty cycle al 60%, ossia l'impulso a livello alto avrà durata pari al 60% dell'intero periodo dell'onda rettangolare generata, mentre l'impulso basso avrà durata (per differenza) pari al 40% dell'intero periodo.
In ogni caso, la lunghezza del periodo non è critica; possiamo decidere arbitrariamente quanto lungo deve essere. L'importante è rispettare le percentuali dei cicli di lavoro. Per il nostro esem-

Figura 8 *Schema elettrico per la verifica del PWM*

ELETTRO@bit

FIERA DELL'ELETTRONICA
INFORMATICA
TELEFONIA
RADIOAMATORIALE
HOBBISTICA
RADIO D'EPOCA

AREZZO
(CENTRO AFFARI)

8-9 OTTOBRE '05

ORARIO
SABATO 9.00 - 19.00
DOMENICA 9.00 - 18.30

ORGANIZZAZIONE GEDIT

TEL. 075.5991028-348.3307007

E-MAIL: GEDIT2004@LIBERO.IT

pio adottiamo una lunghezza di periodo (T) di 100µS (cento microsecondi), per cui il 60% corrisponde a 60 µs ed il 40% corrisponde a 40 µS.

Il software

Questo primo programma è estremamente semplice. Si tratta in pratica di accendere e spegnere, per il tempo sopra calcolato, il piedino RB0. Esaminiamo il relativo software:

```
program PWM01
main:
   trisb=0 'PORTB in Output
   portb=0 'Azzera portb
      while true
      portb.0=1 'RBO ON
      delay_us(60) 'T-ON
      portb.0=0 'RBO OFF
      delay_us(40) 'T-OFF
   wend
end.
```

Verifica

Colleghiamo il tester, in posizione di Volmetro, al punto A dello schema in questione. Leggeremo esattamente 3 Volt. Attenzione però, questa tensione è solo un'illusione "elettrica". Il circuito sta simulando tale differenza di potenziale, semplicemente alternando i periodi di T-ON e T-OFF. Se vogliamo vedere cosa succede "dietro le quinte", occorre collegare un oscilloscopio, sempre al punto A del circuito. Con nostra sorpresa non vedremo una tensione continua di 3 Volt bensì un **treno di impulsi** costanti, con ampiezza 0-5 Volt che, in rapidissima successione, generano il *valore efficace* cercato. Si tratta quindi di una tecnica veramente intelligente che permette di risolvere tanti problemi nel migliore dei modi.

UNA LAMPADA CON QUATTRO GRADI DI LUMINOSITÀ

Ecco un interessante prototipo, che può trovare sicuramente applicazione in un appartamento o un ufficio. Si tratta di una lampada di potenza a 12 Volt, la cui luminosità può essere variata alla pressione di quattro pulsanti n/a. In particolare, il primo pulsante spegne la lampada, il secondo imposta una illuminazione pari al 33%, il terzo imposta una illuminazione pari al 66% ed il quarto la illumina al pieno della sua potenza, quindi al 100%. Realizziamo pertanto lo schema proposto in figura 9.

Analisi procedurale

È il classico circuito a pilotaggio PWM, con cui

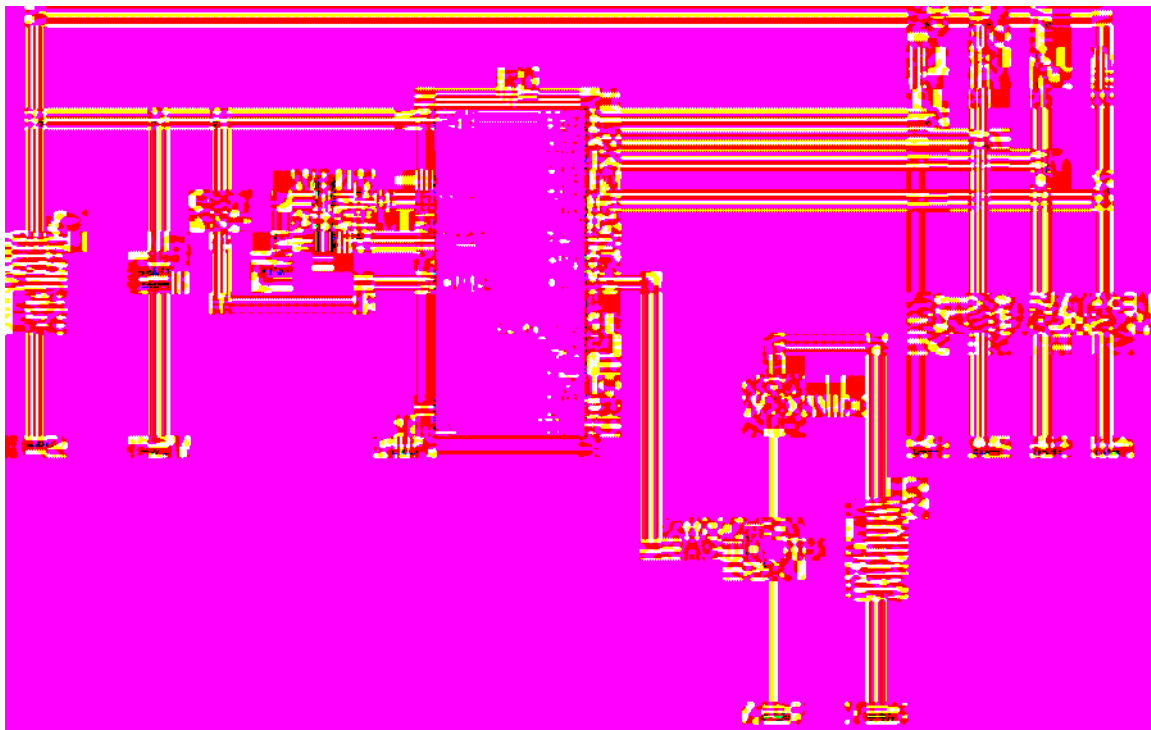



Figura 9 Schema elettrico del variatore di luminosità a PWM

si dosa la luminosità (e quindi la potenza) di una lampada, non agendo sull'intrinseco voltaggio della stessa, ma generando un treno di impulsi proporzionale alla resa desiderata.
Il compito del software è innanzitutto quello di rilevare la pressione di uno dei quattro tasti abilitati. Di conseguenza vengono stabiliti ed assegnati i relativi tempi di duty cycle, ossia di T-ON e T-OFF. Infine viene generata la cadenza relativa ai valori calcolati, per il pilotaggio del transistor di potenza.

Il software

Anche stavolta il software è molto intuitivo. Esaminiamo insieme il listato 1.
Dopo la definizione della funzionalità della PORTB e suo azzeramento, il programma, attraverso gli statements *IF THEN*, controlla lo stato delle quattro porte RB4, RB5, RB6, RB7. A seconda di quale tasto è premuto, avviene l'assegnazione di opportuni valori numerici alle variabili T_ON e T_OFF, che contengono i tempi di attesa per gli stati logici alto e basso. Questi tempi vengono infine attuati dai due loop, che determinano anche la saturazione e l'interdizione del transistor di potenza. Si è scelto di ripetere tante volte il ciclo di attesa, rappresentato dalla funzione *delay_us(1)* in quanto, nella presente release (vers. 1.1.6.0), la suddetta non supporta ancora la parametrizzazione delle variabili. In altre parole non è ancora possibile adottare la seguente clausola: *delay_us(T_ON)*.
Speriamo vivamente che in future versioni questa funzionalità venga implementata.

Collaudo

Non appena si dà tensione al circuito, la lampadina risulta spenta, in quanto inizialmente è stato programmato un valore di duty cycle pari allo 0%. Provando a premere gli altri pulsanti, si noterà la variazione di luminosità della lampada. Consigliamo vivamente di collegare un oscilloscopio alla porta RB0 durante il collaudo, per comprendere come il PIC genera impulsi di diversa larghezza.

LUCE DI CORTESIA

Realizziamo adesso questo prototipo di luce di cortesia, molto utile in automobile.
All'accensione del circuito la lampada resta spenta. Premendo un pulsante, la lampada si illumina e resta in tale condizione per 20 secondi.

[Listato 1]

```
program PWM02
' Variatore di luce a PWM con 4 pulsanti
dim T_ON as byte
dim T_OFF as byte
dim k as byte

main:
   portb=0 'Azzera portb
   trisb=%11110000 'Definisce PORTB
   T_ON=0
   T_OFF=100 'La lampada e' spenta
   while true
      '---Imposta Duty Cicle 0%-----
      if portb.4=1 then
         T_ON=0
         T_OFF=100
      end if
      '--- Imposta Duty Cicle 33%-----
      if portb.5=1 then
         T_ON=33
         T_OFF=67
      end if
      '--- Imposta Duty Cicle 66%-----
      if portb.6=1 then
         T_ON=67
         T_OFF=33
      end if
      '--- Imposta Duty Cicle 100%-----
      if portb.7=1 then
         T_ON=100
         T_OFF=0
      end if
      '----Pilotaggio PWM---
      portb.0=1 'Accende lampada
      for k=1 to T_ON
        delay_us(1) '1us ripetuto n volte
      next k
      portb.0=0 'Spegne lampada
      for k=1 to T_OFF
        delay_us(1) '1us ripetuto n volte
      next k
   wend
end.
```

Trascorso tale periodo si spegne, diminuendo gradatamente e dolcemente (in 10 secondi) la propria potenza senza una brusca interruzione. Realizziamo per lo scopo lo schema di cui alla figura 11.

Il software

Esaminiamo il software riportato nel listato 2.
La temporizzazione degli impulsi avviene grazie alle due variabili t_on e t_off. Quando l'una incrementa, l'altra decrementa, cosicché varia anche la luminosità.

[Listato 2]

```
program PWM03
' Luce di cortesia
dim duty as byte
dim k as byte
dim ripete as byte
dim t_on as byte
dim t_off as byte

main:
   portb=0 'Azzera portb
   trisb=%00000001 'Definisce PORTB
   while true
      if portb.0=1 then 'Se premo pulsante
         portb.2=1 'Accende lampada
         delay_ms(20000) 'Attesa 20 sec.
         for duty=0 to 100
            t_on=100-duty
            t_off=duty
            for ripete=1 to 10
               '-----Avvia PWM-------
               portb.2=1 'Lampada ON
               for k=1 to t_on    'T-ON
                  delay_us(100)
               next k
               portb.2=0 'Lampada OFF
               for k=1 to t_off 'T-OFF
                  delay_us(100)
               next k
            next ripete
         next duty
         portb.2=0 'Lampada OFF
      end if
   wend
end.
```

Al momento della pressione del tasto, la lampada resta illuminata per 20 secondi. Poi gradualmente diminuisce la sua intensità sino a quando si spegne del tutto. Se si vuol cambiare il periodo di spegnimento, occorre modificare il valore del ciclo rappresentato dalla variabile *ripete*.

LUCI NATALIZIE

Molti anni fa, le prime luci elettriche per l'albero di natale lampeggiavano grazie all'intermittente, un circuito primitivo che permetteva alle piccole lampadine di accendersi e spegnersi.
Tale apparecchio generava però, nelle immediate vicinanze, piccoli disturbi elettrici, dovuti proprio alla commutazione interna.
Con il corso degli anni tale circuito venne implementato direttamente nel cordone di alimentazione della treccia luminosa, risparmiando così spazio e costi. L'effetto ottenuto era in ogni caso solamente quello del lampeggio: lampade che si accendevano e spegnevano, senza mezze misure.
Oggi i tempi sono cambiati. Grazie all'adozione di piccolissimi microcontrollori integrati, si è riusciti non solo a programmare il gioco di luci, ossia la particolare sequenza seguita dalle piccole lampadine, ma addirittura a modificarne la luminosità, utilizzando la tecnica del PWM e riuscendo ad ottenere sbalorditivi effetti di transizioni e dissolvenze.
Vediamo come predisporre un progetto che ci consenta di ottenere un gioco di luci natalizio, veramente

Figura 10 Le quattro condizioni della lampada a PWM

Figura 11 *Schema elettrico delle luci di cortesia*

molto grazioso e divertente.

Schema elettrico

Seguite lo schema elettrico della figura 12. Didatticamente abbiamo previsto solo quattro diodi Led collegati rispettivamente alle porte RB0, RB1, RB2 e RB3. Idealmente però potete sostituire tali utilizzatori da trecciole illuminate,

[Listato 3]

```
program PWM04
' Luci di Natale

dim duty as byte 'Percentuale duty cicle
dim k as byte 'Variabile di comodo
dim t_on as byte 'Tempo in ON
dim t_off as byte 'Tempo in OFF
dim i as byte 'Variabile di comodo
'Array dei Giochi
const led as byte[10]=(1,2,4,8,3,6,12,7,14,15)

main:
   portb=0 'Azzera portb
   trisb=0 'PORTB in OUTPUT
   while true 'Ciclo infinito
      for i=0 to 9
         '----Accensione Graduale
         for duty=1 to 100
            t_on=duty
            t_off=100-duty
            '-----Avvia PWM-------
            portb=led[i] 'Led ON
            for k=1 to t_on    'T-ON
               delay_us(50)
            next k
            portb=0 'Led OFF
            for k=1 to t_off 'T-OFF
               delay_us(50)
            next k
         next duty
         '----Spegnimento Graduale
         for duty=100 to 1 step -1
            t_on=duty
            t_off=100-duty
            '-----Avvia PWM-------
            portb=led[i] 'Led ON
            for k=1 to t_on    'T-ON
               delay_us(50)
            next k
            portb=0 'Led OFF
            for k=1 to t_off 'T-OFF
               delay_us(50)
            next k
         next duty
      next i
   wend
end.
```

composte da tante lampade, previo pilotaggio da parte di opportuni transistor.

Il Software

Esaminiamo il programma, riportato nel listato 3, passando successivamente ai commenti.
I vari giochi di luce sono memorizzati nel vettore *led[i]*, permettendo un riutilizzo del codice sottostante. Si tratta in sostanza di una serie di cicli nidificati, che modificano proporzionalmente il duty cycle degli impulsi erogati ai diodi led, consentendone la variazione luminosa con continuità.

PILOTIAMO UN MOTORE

La tecnica del PWM si presta egregiamente a pilotare un motore elettrico consentendone la variazione di velocità. Due sono gli aspetti che maggiormente spiccano usando tale metodo:

1. Si riduce la dissipazione di calore poiché i transistor sono utilizzati come interruttori e non come amplificatori.
2. Anche a bassi regimi il motore conserva un'elevata coppia, dal momento che, anche con duty cycle minimo, esso è sempre alimentato con la tensione massima di alimentazione.

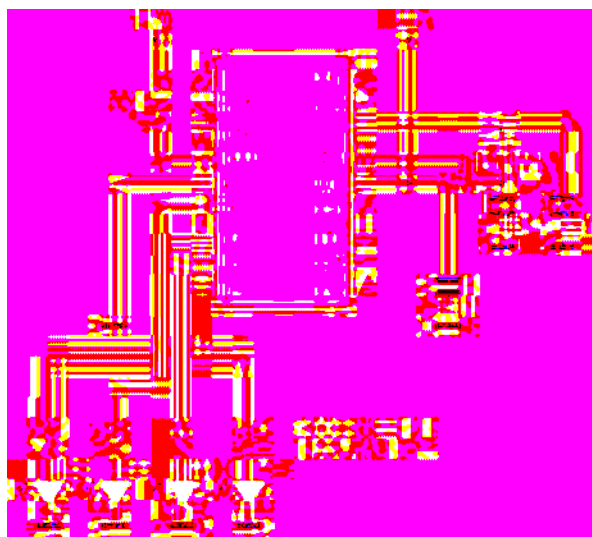



Figura 12 *Schema elettrico delle luci di Natale*

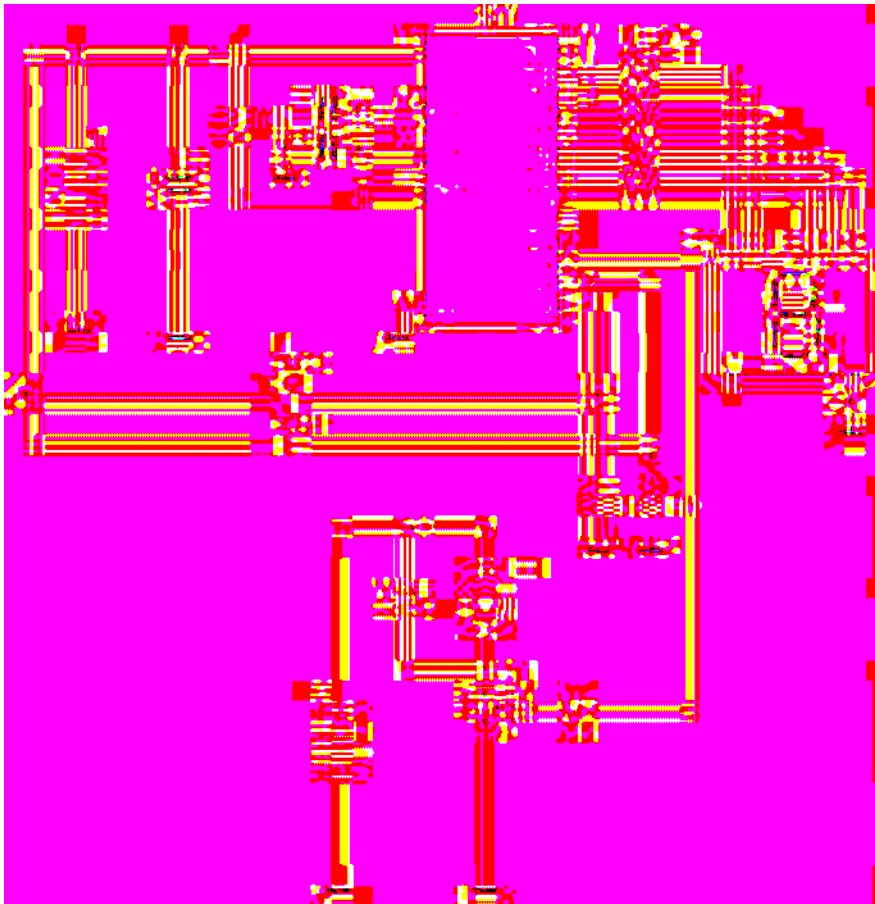



Figura 13 *Schema elettrico del variatore per motore*

Il nostro progetto

È sicuramente uno dei più interessanti progetti proposti nel nostro corso. Occorre realizzare un circuito dotato di due pulsanti ed un motore. I pulsanti sono utilizzati a mo' di cambio marce. La pressione del primo tasto aumenta la velocità del motore mentre il secondo tasto diminuisce tale velocità.
Il sistema prevede dieci velocità diverse, ossia dieci step che fanno variare il regime del motore da un minimo (fermo) ad un massimo. Il passo impostato è visualizzato attraverso un display a 7 segmenti, che in tal modo mostra i numeri da 0 a 9.

Schema elettrico

Lo schema, visibile in figura 13, riporta i componenti fondamentali del progetto. La PORT-B è configurata completamente in output e serve esclusivamente a pilotare il display a 7 segmenti.
La PORT-A ha invece un funzionamento promiscuo, in quanto i pin RA0 e RA1 fungono da ingressi per i due pulsanti, mentre il pin RA2 pilota con impulsi regolari il motore, attraverso il mosfet di potenza IRF540 o equivalente. Il diodo in controfase collegato in parallelo al motore elimina le extratensioni, potenzialmente pericolose per il transistor. È possibile in ogni caso collegare una grossa lampada al posto del motore.

Il software

Un piccolo gioiello di programmazione, che in poche righe riesce ad eseguire tante funzionalità. Si inizia con la definizione delle variabili di lavoro e del vettore contenente le codifiche per la visualizzazione dei dieci numeri sul display. Si passa quindi alla definizione delle porte di comunicazione e relativo azzeramento.
Vitale importanza riveste il flag *premuto*, che ha il compito di stabilire se un tasto è stato appunto premuto o meno.
Questa variabile consente il controllo della routine di incremento e decremento, evitando il fastidioso effetto del *rimbalzo* e del *repeat*. Il valore acquisito dallo *step* è così assegnato alla variabile *marcia* che, assieme alle variabili *t_on* e *t_off*, si occupano del pilotaggio del motore attraverso il dosaggio della larghezza dell'impul-

Figura 14 *Funzioni svolte dalla PORT-A e dalla PORT-B*

È in Edicola fino a fine Settembre

da NON PERDERE!!

[Listato 4]

```
program PWM05  ' Motore

   const display as byte[10]= (63,6,91,79,102,109,125,7,127,111)
   const test as byte[7]=(1,2,4,8,16,32,64)
   dim marcia as byte 'Contiene la marcia impostata
   dim premuto as byte 'Flag che indica tasto premuto
   dim k as byte 'Variabile di comodo
   dim t_on as byte 'Durata impulso ON
   dim t_off as byte 'Durata impulso OFF

main:
   porta=0 'Azzera porta
   portb=0 'Azzera portb
   trisa=00011 'Definisce port-a
   trisb=0 'definisce PORTB in output
   marcia=0 'Imposta la marcia 0 (motore fermo)
   t_on=0 'Durata inpulso ON iniziale
   t_off=90 'Durata inpulso OFF iniziale
   premuto=0 'Si suppone che non si preme nulla
   while true
      portb=display[marcia] 'Visualizza marcia
      '----Antirimbalzo e antirepeat------
      if (porta.0=0) and (porta.1=0) then
         premuto=0 'I tasti non sono premuti
      end if
      '----Se si preme tasto incremento--------
      if (porta.1=1) and (marcia<9) and (premuto=0) then
         marcia=marcia+1
         t_on=t_on+10
         t_off=t_off-10
         premuto=1
      end if
      '----Se si preme tasto decremento--------
      if (porta.0=1) and (marcia>0) and (premuto=0) then
         marcia=marcia-1
         t_on=t_on-10
         t_off=t_off+10
         premuto=1
      end if
      '--PILOTA IL MOTORE CON IL PWM-----
      porta.2=1 'Impulso ON
      for k=1 to t_on
         delay_us(1)
      next k
      porta.2=0 'Impulso OFF
      for k=1 to t_off
         delay_us(1)
      next k
   wend
end.
```

so fornito. Studiatelo a fondo e cercate di comprendere bene tutte le routine utilizzate, in quanto si tratta di uno dei primi software aperti, e costituisce la base iniziale per la realizzazione di prototipi professionali ed industriali *[Listato 4]*.
In figura 14 riportiamo la funzione svolta da ciascun piedino delle porte A e B.

Raccomandazioni per l'uso dei motori

Di seguito elenchiamo alcune piccole precauzioni da prendere durante l'utilizzo dei motori, onde evitare eventuali problemi tecnici:

- Usate sempre il diodo di blocco posto in antiparallelo al motore per impedire la formazione di pericolose extratensioni.
- Dotate sempre il transistor di una adeguata aletta di raffreddamento.
- Non bloccate MAI manualmente l'attività del motore, pena l'immediata distruzione del transistor.
- Utilizzate una frequenza di PWM sempre superiore a 20 Khz: in caso contrario potrebbe udirsi all'interno del motore stesso una nota acustica di frequenza equivalente al periodo calcolato.

CONCLUSIONI

Bene, purtroppo anche questa puntata è terminata. Speriamo di essere riusciti a dare gli spunti necessari alla realizzazione di utili apparecchiature.
Ricordate sempre che il miglior modo per imparare a programmare con disinvoltura è quello di provare e riprovare ancora. Solamente con la costanza e con la perseveranza si potranno acquisire alla perfezione molti metodi e trucchi della programmazione.

Vi diamo appuntamento alle prossime puntate, ricche di novità e di nuove idee, non prima però di avervi augurato un buon lavoro.

Economica CNC

È facile trovare in rete diversi progetti di CNC ma ben poche sono, economicamente parlando, pensate per l'hobbista. Con questo chiodo in testa ho voluto vedere cosa si riusciva a fare spendendo poche decine di Euro ed i risultati sono davvero interessanti...

Prima di costruire una macchina CNC, per prima cosa dobbiamo dividere il problema nei suoi aspetti principali, in particolare:

1. **Parte meccanica**. È il vero punto nodale, più siamo bravi qui, più la nostra macchina sarà precisa rispondendo alle nostre esigenze.
2. **Parte elettronica**. Qui con uno schema giusto si può risolvere in fretta... non è proprio così, dipenderà molto dal software che vogliamo usare, vi sono infatti diversi modi per inviare segnali ai motori.

Figura 1 Vista d'insieme della CNC ultimata

3. **Software**. Qualcosa freeware si trova, non siamo a livelli top ma almeno nel mio caso fanno di più di quanto non riesca a fare la meccanica.

PARTE MECCANICA

Nel mio caso qui sono cominciati una marea di problemi, volendo spendere poco, era necessario trovare alternative "decenti" a quanto di dedicato ci fosse in commercio.
Girovagando in rete mi sono fatto un'idea dei vari schemi realizzati; in particolare ho notato che sono tre le realizzazioni più comuni:

1. Una realizzazione (tipo pantografo) prevede che il pezzo da lavorare sia fermo ed è l'utensile a muoversi sui tre assi.
2. Una struttura a ponte fisso: il pezzo si muove lungo l'asse X mentre sul ponte il carrello porta utensile si muove nelle direzioni Y e Z.
3. Altra soluzione con il tavolo a croce, il pezzo si muove lungo X e Y (su un tavolo a croce) e l'utensile ha solo il compito di alzarsi e abbassarsi.

Non mi chiedete perché ma io mi sono lanciato sul secondo tipo, quello a ponte, il risultato finale è quello mostrato nella figura 1.

L'unico modo per procedere è stato quello di concentrarmi sulle singole parti, cominciando dal piano d'appoggio e l'asse X, vediamo in dettaglio.
Come noterete è in legno, giusto per darvi un idea la base è una tavola di truciolare rivestita in formica (la trovate nei centri di bricolage dove volendo la tagliano anche secondo le vostre misure) spessore 18 mm e 800x500 mm in dimensione, lo vedete in primo piano nella figura 2. Su questo piano va fissato l'asse X che ovviamente scorre... primo problema.
Ho trovato delle guide per mobili che vi mostro

a 3 assi

di Marco Fabbri
marnic@roboitalia.com

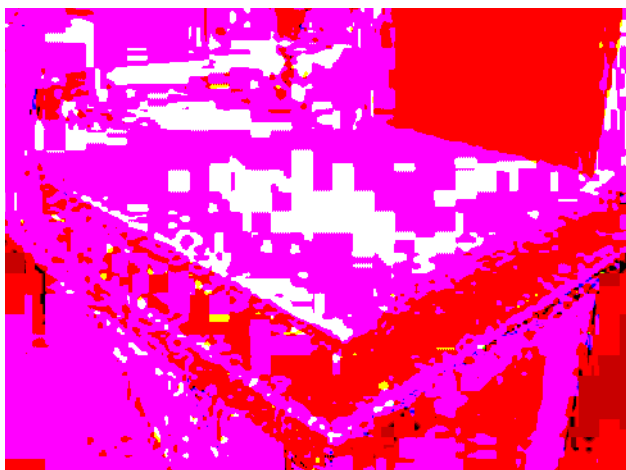



Figura 2 Piano base e piano asse X

nella figura 3, scorrono abbastanza bene e si muovono su sfere.

Quelle per l'asse X sono lunghe 45 cm e hanno una corsa di quasi 70 cm ma egli estremi sono un po' ballerine, sono bene utilizzabili per 50, 55 cm di corsa utile, sopra queste guide ho fissato il piano X che sarà poi quello dove fissare il pezzo in lavorazione, questo piano nel mio caso è di 45x40cm. È importante fissare le guide in modo che siano parallele, le guide vanno fissate alla base di appoggio ad una distanza tale da poter contenere il piano X più un po' di spazio per qualche vite, seguendo le mie misure, le mie sono fissate a 36 cm (distanza tra gli assi delle guide) è anche molto importante una buona perpendicolarità con il lato

Figura 3 Particolare modello delle guide usate per gli assi

Figura 4 Particolare chiocciola

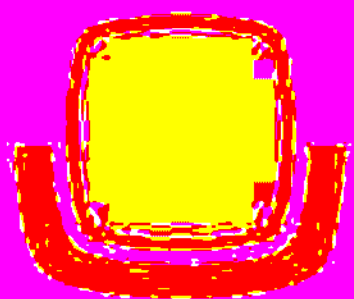
dove verrà fissato il motore.
Bene ora posizioniamo il piano X e segniamo dove forare il legno per fissarlo alla parte mobile delle guide, forate, verificate che il tutto scorra ancora liberamente, se all'inizio o alla fine della corsa scorre con difficoltà significa che le guide non sono parallele, sistematele agendo sul gioco che hanno le viti che fissano le guide al piano di appoggio, se siete sfortunati e non basta....
Ora dobbiamo trovare il sistema di trasmettere il moto dal motore al piano X. Io ho preso sempre nel solito centro di bricolage, delle barre filettate in ottone da 6 mm (filetto M6). Cercate di scegliere quelle più diritte possibile, la linearità non è importante per la precisione, l'errore sarebbe di pochi micron, ma è importante per ridurre tensioni e sforzi sul motore o sul supporto reggispinta. Ci sono anche barre in ferro, sono più difficili da tagliare e sono meno flessibili, caratteristica che ci servirà per correggere la mancanza di linearità.
Visto che siete al reparto dove vendono le barre filettate, prendete anche un metro di profilato pieno a sezione quadrata di alluminio da almeno 10 mm di lato, ci serve per fare la chiocciola. Non ho trovato infatti un sistema migliore già pronto.
Tagliate un pezzo di profilato di circa 6-7 cm di lunghezza e foratelo con una punta da 3 mm per realizzare i fori di fissaggio. Forate sull'altro lato con una punta da 5.5mm filettate quindi il foro con un maschio da M6.
Per questi fori è importante la perpendicolarità, vi consiglio l'uso di un trapano a colonna, procedete con cautela, è un pezzo importante!

Fissate la chiocciola appena realizzata sotto il piano X, deve essere assolutamente perpendicolare alle guide, se poi come nel mio caso il profilato dal quale avete ricavato la chiocciola è da 10mm sarà necessario distanziare la chioc-

ciola dal piano, diversamente la barra filettata striscerà sul piano, per fare questo interponete uno o due dadi come si intravede nella figura 4. Visto che questo pezzo va fissato al piano X di legno, io ho usato delle viti M3 a testa svasata, in questo modo serrando bene il piano rimane liscio.
Bene, ora tagliamo la barra filettata, nel mio caso circa 70cm, fissiamo il piano X alle guide e avvitiamo la barra alla chiocciola, noterete che può ondeggiare (e questo ci aiuterà) sarà invece praticamente assente il backlash se avete eseguito il filetto con cura.
Ora dobbiamo fissare un lato della barra al piano di appoggio, ecco come: ho trovato delle squadrette già forate come visibile in figura 5 e (un po' di fortuna non guasta) di misura giusta. Il foro dove passa la barra filettata deve comunque essere ad un'altezza tale da far si che la barra sia orizzontale e parallela al piano di appoggio.
Come vedete ho utilizzato dei cuscinetti, uno per lato stretti da dado e controdado da una parte e dall'altra della squadretta, in questo modo non ho attriti di rotazione e nemmeno sollecitazioni meccaniche sul motore.

Figura 5 Squadretta per fissare la barra filettata

Premetto che due dei tre motori li ho comprati in una delle tante fiere di elettronica per 5 Euro cadauno, hanno 200 passi/giro e purtroppo una resistenza piccola 5 ohm, funzionano a 5 V e si "bevono" 1 Ampere (avrò l'elettronica sotto stress).
Fissare il motore al piano di appoggio alla misura giusta è solo questione di un po' di precisione e manualità, una semplice piattina di alluminio forata e piegata vi darà il risultato che cercate (magari dopo qualche prova andata male).
Inutile dire che asse motore e asse barra filettata devono essere sulla stessa retta.

Figura 6 Morsetti "mammuth" dove troviamo i giunti

Non ci rimane che accoppiare motore e barra… e qui possono essere dolori.
Ve li ricordate i vecchi "mammuth"?
Sono quelli nella foto, in particolare quelli bianchi (in foto, non è la regola) sono da 16 mmq e i neri da 25 mmq.

Dentro trovate due manicotti con fori filettati, quello da 16 mmq è giusto per la barra filettata che abbiamo usato! Tanta fortuna? In effetti pareggiamo con i motori, è facile che (come nel mio caso) troviate motori con albero da 1/4 di pollice equivalente a circa 6,35 mm… una buona lima e tanta pazienza, un po' al manicotto e un po' all'albero motore, limando il manicotto la barra avrà più gioco e successivamente stringendola con la vite si disasserà rispetto al motore. Una soluzione buona da non credere è fare qualche giro di nastro isolante attorno alla barra (solo la parte da inserire nel manicotto ovviamente) per riportarla a misura giusta.
Un accoppiamento simile è molto rigido quindi l'inevitabile errore di disassamento farà

Figura 7 Fissaggio ancoraggio motore

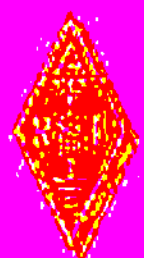



Figura 8 Pannello per fissaggio asse Z

ondeggiare il motore, a parte una perdita di potenza e un po' di rumore non vi sono altri inconvenienti, ciò non contribuirà a nessun errore. Altra soluzione è comprare dei giunti, costano molto e io li ho trovati solo da RS, sono visibili nella figura 5.
Devo dire che il motore non balla più ma le prestazioni complessive non ne hanno tratto vantaggio. Potete anche notare la semplicità delle staffe per il motore.
Passiamo ora alla struttura a ponte.
Non è facile da raccontare e spero si capisca qualcosa dalle foto, figure 7, 8, 9. Con del legno ho realizzato un vero e proprio ponte che attraversa il piano X, barra filettata, chiocciola e ancoraggio della barra sono identici come concetto, unica eccezione sono le guide per i cassetti che hanno una lunghezza di 35 cm per l'asse Y e 28 cm per l'asse Z

Qui è veramente importante che il ponte si muova perpendicolarmente alle guide dell'asse X, se pensate di forare circuiti stampati o altro prestate molta attenzione a questo punto.
Nelle figure si vede dove sono le guide, ad esse è fissato un pannello di legno da 30 cm per 30 cm che sarà da supporto al carrello per l'asse Z.

Figura 9 Struttura ponte, vista d'insieme

Anche per l'asse Y valgono le stesse attenzioni, guide parallele e barra filettata parallela alle guide, abbiamo detto della perpendicolarità con l'asse X ma è anche molto importante che l'asse Y sia orizzontale in modo che l'utensile sia alla stessa altezza dal piano di lavoro sia con asse Y al minimo che al massimo. Altro punto è tenere in piedi il ponte e tenerlo verticale, le particolari squadrette che vedete in figura 7 forniscono un'ottima rigidità alla struttura e le staffe ad L alla base del ponte completano l'opera.
Passiamo all'asse Z.
Realizzato sempre in legno, le guide come per l'asse Y, barra e chiocciola al solito modo, vediamo qualche foto, figure 10, 11, 12.

Le immagini dovrebbero parlare da sole e se avete realizzato il resto questo non sarà un problema. Dovrete solo sagomare due pezzi di legno in modo che stringendo le viti il trapano sia bloccato. Inutile dire che l'asse Z deve essere perfettamente verticale.
Bene, dichiarerei chiusa la parte meccanica, siete un po' increduli sulla solidità del tutto?

Figura 10 Vista generale asse Z

Figura 11 Fissaggio fresa

Figura 12 Fissaggio asse Z con asse Y

Non avete tutti i torti, non sarà una macchina professionale ma fino a questo momento se escludiamo i motori abbiamo speso circa 30 Euro, per i motori spero abbiate trovato qualche vecchia stampante ad aghi da sezionare, di solito contengono dei motori fantastici.

PARTE ELETTRONICA

Il comando dei motori (che inevitabilmente sono stepper) è demandato ad una scheda elettronica che non è difficile da realizzare.
Vi propongo uno schema per iniziare, dicendo subito però che vi sono programmi che utilizzano altri standard di segnali sulla parallela. Per esempio i programmi che gestiscono un motore alla volta, usano sempre quattro pin per la sequenza delle fasi usano poi un pin per ogni asse (3 nel nostro caso) per dare il segnale enable al motore, si potrebbe obiettare che movendo un motore per volta viene persa precisione, è vero ma solo in teoria, esempio: nel nostro caso facciamo fare uno step (anzi mezzo) all'asse X e dopo all'asse Y ne consegue che avrò disegnato un gradino e non una linea… il nostro gradino però ha il lato di 400 mezzi passi per fare un giro motore, 1 mm è l'avanzamento del nostro asse con barra M6 per ogni giro per cui avremo un errore massi-

Elenco componenti

Sigla	Valore
C1	3,3 nF ceramico
C2	100 nF poliestere
D1-D8	Diodi Schottky
U1	L297
U2	L298
R1	22 KΩ 1/4 W

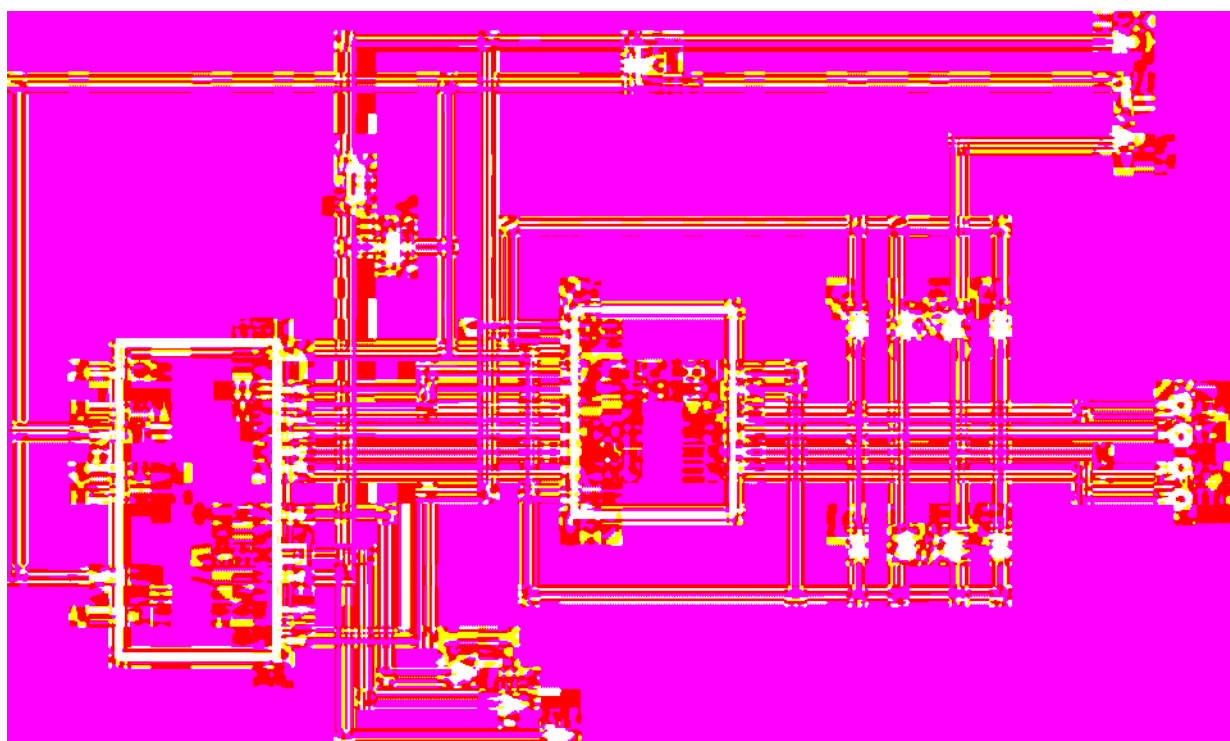



Figura 13 Schema elettrico interfaccia PC

mo di 1/400mm lato dello scalino che, dopo l'opportuna geometria, da 0,00177 mm di errore massimo (se cercate di meglio non costruite il mio prototipo).
Il vantaggio di questo sistema di comando è che per aggiungere un asse in più serve solo un pin in più, per comandare sei assi servono 4+6 pin della parallela

Altro sistema utilizzato è di usare 4 pin per ogni motore e quindi comandare veramente in parallelo i nostri motori, ci serviranno però per sei assi 6*4=24 pin.

Per essere però coerenti con quanto sarà presentato di seguito direi di analizzare (e realizzare) l'interfaccia che prevede di comandare i motori nel modo "STEP/DIR", questo significa che utilizzerò due pin per ogni motore, il primo pin invierà al motore un impulso per ogni passo che vogliamo faccia il motore, il secondo pin sarà quello che determina con il suo stato, alto o basso, il senso di rotazione del motore.
Lo schema raffigurato in figura 13 è relativo al questo ultimo caso, lo schema è relativo ad un solo motore o asse, ne dovrete pertanto realizzare tre nel caso di una CNC a 3 assi.

Come vedete viene usata la coppia di integrati L297-L298 e lo schema è esattamente quello riportato sul datasheet di entrambi i componenti dove oltretutto viene proposta una soluzione di circuito stampato, se non trovate il datasheet cercate su: www.alldatasheet.com vi assicuro che c'è. Questa coppia di integrati risolve molto i problemi di pilotaggio dei motori stepper, serve infatti solo fornire i due segnali di passo e direzione perché il 297 attivi in sequenza i giusti rami dei ponti ad H contenuti nel 298.
Realizzati i tre gruppi non ci resta che collegarli ad una presa a 25 poli, per ogni gruppo motore ci servono solo gli ingressi STEP, DIR e Enable, questi ultimi saranno collegati insieme e portati ad un unico pin della presa, i rimanenti a scelta su uno dei pin di uscita dalla presa ad esempio come riportato nella tabella seguente.

Le uscite del L298 vanno ai 4 fili che escono dai motori passo-passo bipolari e se non siete particolarmente fortunati non beccherete la sequenza giusta, per aiutarvi identificate i due avvolgimenti, le uscite OUT1 e 2 vanno ad un avvolgimento OUT 3 e 4 all'altro, se il motore non gira invertite i fili di uno solo dei due avvolgimenti, tutto dovrebbe essere a posto ora.

Esempio collegamento assi		
Asse	**Pin STEP**	**Pin DIR**
X	2	3
Y	4	5
Z	6	7

Portate ora 4 fili dai pin di ingresso della presa a 25 poli ad esempio i pin 10, 11, 12, 13 e un filo dal pin 24 di massa, qui collegheremo i tre finecorsa degli assi più un pulsante che sarà adibito alla funzione di palpatore dell'utensile, non ha importanza se userete contatti aperti o chiusi, vedremo come configurarli correttamente.
Due parole sul cavo che collegherà la LPT del nostro PC alla nostra interfaccia, in commercio non è difficile trovare dei cavi pin-to-pin ossia i pin corrispondono sulle prese dove si attesta il cavo, volendo potete costruirvelo rispettando le corrispondenze 1 con 1, 2 con 2, ecc.

PARTE SOFTWARE

Abbiamo costruito una macchina in grado di lavorare prendendo informazioni dal computer, è quindi ovvio che ci serve un programma che partendo dal nostro disegno o progetto traduca il tutto in sequenze di impulsi sulla porta del PC dove abbiamo collegato la nostra macchina.
Anche se ormai la porta USB si sta rivelando adatta un po' a tutto, ancora oggi e specialmente in realizzazioni hobbistiche la cara e vecchia porta parallela è la più usata per queste applicazioni.
Il vantaggio di avere sui tanti pin disponibili delle tensioni a 5V ci permette di realizzare in modo semplice un'interfaccia per muovere i nostri motori. I pin di input, sono quelli che dicono al PC che la stampante ha finito la carta, che non è in linea, eccetera. In questo modo il tutto si trasforma in una semplice logica combinatoria.
Procediamo con ordine: la nostra LPT1 ha

come sappiamo 25 pin di cui 12 di uscita e 5 di ingresso come vedete dalla tabella 1, i pin di uscita sono usati per comandare la CNC, far girare gli stepper, azionare la fresa e altre funzioni ausiliarie che potrebbero essere ad esempio un refrigerante o una aspirazione per i trucioli; gli ingressi sono sostanzialmente usati per i finecorsa, il palpatore e l'arresto d'emergenza.

Quasi tutti i software danno la possibilità di configurare i vari pin e quindi adattare la vostra interfaccia, inoltre il software può gestire, per i pin di input, sia il contatto normalmente chiuso che quello normalmente aperto.
In questo articolo prenderemo in considerazione un software particolarmente semplice da usare e piuttosto completo, caratterizzato da vari moduli che ci permettono di disegnare e realizzare il nostro pezzo oltre che a importare altri formati grafici tra cui ovviamente l'ISO GCode ma anche il semplice bitmap; sto parlando di **Ninos** (www.cadcam-ninos.com) giunto oggi alla versione 2.11 (vedi figura 14).
Dal sito è scaricabile una versione di valutazione i cui limiti sono nel modulo di lavorazione che non permette di processare più di un certo numero di vettori, vi rimando al sito per tutte le informazioni riguardanti licenze e quanto altro relativo ai copyright.
Lanciando Ninos si apre una piccola finestra dalla quale si può accedere ai diversi moduli, vediamoli brevemente:

Modulo tornitura

È un modulo relativo a macchine automatiche di tornitura, la nostra CNC non ci permette di usare questo modulo e molto sinceramente non l'ho guardato più di tanto, però chissà, magari a qualcuno viene voglia di farlo.

Modulo CAC per disegno 2D 1/2

È un CAD molto semplice e dedicato alla lavorazione, non è definibile come un 3D in quanto le singole primitive (cerchi, rette, punti, poligoni ecc.) possono avere una singola profondità, per chiarire il concetto, posso realiz-

Pin5	Segnale	Direzione	Descrizione
1	Strobe	PC > Stampante	Attivazione lettura dati
2	Data 0	PC > Stampante	1° bit del dato
3	Data 1	PC > Stampante	2° bit del dato
4	Data 2	PC > Stampante	3° bit del dato
5	Data 3	PC > Stampante	4° bit del dato
6	Data 4	PC > Stampante	5° bit del dato
7	Data 5	PC > Stampante	6° bit del dato
8	Data 6	PC > Stampante	7° bit del dato
9	Data 7	PC > Stampante	8° bit del dato
10	ACKNLG	Stampante > PC	Segnale di inizio ricezione
11	BUSY	Stampante > PC	Segnale di fine ricezione
12	PE	Stampante > PC	Segnale di fine carta
13	SLCT	Stampante > PC	Collegamento a +5 volt
14	AUTO FEED	PC > Stampante	Avanzamento carta dopo CR
15	ERROR	Stampante > PC	Errore
16	INIT	PC > Stampante	Inizializzazione stampante
17	SLCT IN	PC > Stampante	Attivazione PRN con codice DC
18-24	GND		Massa

***Tabella 1** Piedinatura e direzione dei segnali della porta parallela LPT*

zare un cerchio di profondità 2 e un rettangolo di profondità 3 ma non posso realizzare una retta che partendo da profondità 2 termini a profondità 3. Quello che ne risulta è che se devo realizzare pezzi anche particolari e difficili ma con tutte le facce parallele all'asse x o y non ci sono problemi, difficile o quasi impossibile realizzare piani inclinati o curve funzioni di Z. A parte questo limite, per ogni primitiva posso scegliere come questa verrà lavorata, se è un contorno interno od esterno, uno svuotamento o lavorazioni laterali secondo le funzioni G41 o G42 (Gcode), l'ordine di lavorazione, l'utensile utilizzato e le caratteristiche della lavorazione (velocità, numero di passate ecc.).
Anche da questo modulo si può accedere al magazzino utensili, dove si possono configurare un gran numero di frese e punte, definirne le dimensioni e la lavorazione alla quale sono destinate, si può associare poi un immagine bmp in modo da rendere visibile il tipo selezionato. Due parole sui formati supportati, il modulo può importare i formati DXF a due dimensioni, BMP e HPGL

Figura 14 *Schermata di accesso a Ninos, attorno al pulsante verde per la lavorazione ci sono i vari moduli in funzione dell'origine del vostro file*

Figura 15 *Particolare del magazzino utensili dove è possibile impostare tutte le caratteristiche e i parametri massimi di lavorazione*

Modulo BMP > 3D

Questo è un modulo interessante se si vogliono realizzare lavorazioni partendo da file immagine tipo BMP, GIF, JPG e WMF, in pratica vengono realizzati piani diversi in funzione delle caratteristiche grafiche tipo luminosità colore ecc. modificando questi parametri potete vedere in diretta come cambia il risultato 3D, è un modulo molto immediato nella comprensione e nel suo utilizzo basta ciccare qualche bottone per vederne l'effetto.

Modulo incisore CS

Per noi appassionati di elettronica, questo modulo, l'ultimo arrivato nello sviluppo di Ninos, è ancora piuttosto incompleto, ha ancora qualche baco e manca ancora di buona parte della traduzione in italiano, viene distribuito insieme a due sottomoduli, anche questi incompleti e un po' rudimentali ma se pensate che funzionano senza limiti non potrete che apprezzare, i due sottomoduli servono a disegnare schemi elettronici e hanno anche una libreria di componenti che permette di fare quasi tutto ciò che serve, da questo si passa al secondo sottomodulo che vi permette di creare le tracce del circuito stampato, comprende anche una funzione di sbroglio automatico, in pratica se non abbiamo esigenze professionali direi che si possono usare anche senza poi avere una CNC con la quale incidere il circuito stampato, il modulo di incisione vero e proprio oltre a lavorare i file che abbiamo appena generato può caricare file di tipo Gerber che è lo standard con cui salvano quasi tutti i software anche professionali. Molte cose sono ancora un po' troppo... in francese per i miei gusti per cui non aggiungo altro e attendo la versione tradotta.

Modulo STL > 3D

Moltissimi software di grafica tridimensionale tipo solidworks, rhinoceros e altri possono

esportare in formato STL, il modulo ne esegue una vera e propria scansione grafica che poi viene diviso in due fasi di lavorazione, una fase di sgrossatura dove in funzione dei parametri che abbiamo impostato, passata dopo passata verrà "scolpito" il nostro pezzo mantenendo un margine di abbondanza che possiamo definire noi per la fase successiva di finitura per la quale possiamo ovviamente cambiare utensile e velocità. La finitura può essere poi eseguita sia per scansioni sull'asse X che per scansioni sull'asse Y. E' importante dire che questo modulo produce lavorazioni più lunghe come tempi di esecuzione rispetto alla lavorazione per primitive, si parte infatti sempre da un parallelepipedo che contiene il pezzo da "scolpire" per cui verranno lavorate completamente anche quelle parti in eccesso che sarebbe molto più rapido tagliare con una sola lavorazione, il vantaggio è quello di poter ottenere profili particolari e attraverso le opzioni disponibili lavorare in negativo cioè ottenere lo stampo del nostro pezzo che può essere suddiviso in varie altezze così da ottenere stampi contenenti dei sottosquadri impossibili da ottenere se non con macchine molto complesse. Vi assicuro che le potenzialità di questo modulo sono veramente la soluzione a molti problemi.

Come potete notare tutti questi moduli fanno poi riferimento al software di lavorazione vero e proprio rappresentato dal pulsante verde al centro, è qui che impostiamo la nostra macchina, vediamolo: la prima cosa da fare è configurare la nostra interfaccia e dire quindi a Ninos quali pin della parallela azionare per muovere i nostri motori, per fare questo andiamo nel menu Lavorazione, quindi configurazione e facciamo click su comunicazione LPT, la maschera che ci appare, figura 16, è divisa in quattro parti, in alto a sinistra possiamo scegliere quale porta utilizzare ed eventualmente se usare solo il simulatore, l'avanzamento dell'utensile per ogni asse per ogni singolo passo dello stepper, il campo accetta anche le frazioni per cui se avete un motore da 200 passi per giro e un avanzamento di 1mm per giro potete scrivere direttamente 1/200, a scanso di equivoci c'è comunque un pulsante che richiama la calcolatrice. Subito sotto trovate i campi dove inserire la velocità massima di avanzamento di ogni asse, attenzione questo non ha nulla a che vedere con la velocità di lavorazione ma solo con i limiti della macchina, dei motori nonché del PC che in funzione della sua velocità e del sistema operativo e dei task attivi riuscirà a comunicare con la LPT con una certa frequenza; esiste un pulsante "Calibrare LPT" che verifica appunto che il PC sia in grado della giusta velocità, in caso contrario è necessario ridurre la velocità massima oppure chiudere gli altri task attivi e anche come ultima risorsa provate a disabilitare la visualizzazione in tempo reale dei contatori e dell'indicatore di velocità, trovate queste opzioni nel menu Lavorazione/configurazione/Opzioni nel tab indicatori.

Sempre in alto ma a destra trovate un grafico e delle barre di scorrimento dove poter regolare le accelerazioni, questo è molto utile per capire i limiti dei motori che non hanno evidentemente la possibilità di arrivare alla velocità massima partendo da fermi in un tempo zero.
Passiamo sotto nella sezione OUT per configurare finalmente i pin della nostra interfaccia, per ogni asse si deve selezionare il pin dove vengono inviati gli impulsi relativi ai passi e il pin che determina il senso di rotazione. Nella colonna di destra della sezione OUT abbiamo la configurazione delle funzioni ausiliarie, di questi, come vedete dalla mia configurazione io utilizzo il pin 8 che mi eccita il relé che fa partire la fresa e il pin 17 Alimentazione moto-

Figura 16 *La maschera di configurazione, notate il disegno del connettore LPT per riconoscere i pin*

re, quest'ultimo rappresenta l'enable della mia interfaccia, in sostanza questo pin viene abilitato prima di mandare i clock ai motori degli assi e viene disabilitato a lavorazione finita o durante una pausa per non tenere sotto tensione gli stepper inutilmente. Passiamo alla sezione destra, "IN" dove possiamo impostare i pin relativi ai segnali che vengono dalla nostra CNC, in particolare i tre finecorsa, un palpatore e due stop di emergenza.
Il palpatore, non è un particolare tipo di maniaco, bensì un pulsante messo in un punto definito del piano di lavoro e ad una ben precisa altezza dal piano di lavoro, a cosa serve? Tutte le volte che cambiate utensile lo stringerete nel mandrino avrete una diversa altezza tra la punta della fresa e il pezzo da lavorare, per ricalibrare il tutto basterà nel menu utensile chiedere di palpare l'utensile, la macchina si porterà alle coordinate XY dove avete indicato essere il palpatore e comincerà a far scendere la punta fino a quando non tocca il palpatore del quale conosciamo l'altezza, è facile quindi intuire che è uno strumento di estrema utilità.
Una volta impostato il tutto con il pulsante "Provare" potrete verificare muovendo i motori, azionando i finecorsa che macchina e software si comprendano e si scambino le informazioni corrette. Possiamo ora passare al menu di configurazione definizione degli indicatori, qui si tratta solo di indicare dei parametri macchina che sono piuttosto intuitivi, è anche la maschera dove indicare le coordinate del palpatore di cui abbiamo parlato, andiamo oltre al menu di configurazione opzioni, nella linguetta opzioni dobbiamo scegliere se la nostra macchina ha o meno i finecorsa, se verificare il flusso dati sulla LPT ad ogni avvio, se fermare il mandrino nella fase in cui ci si sposta da un oggetto e il successivo, in questo caso è addirittura possibile definire una temporizzazione per dare tempo alla fresa di andare a regime di rotazione.
Su limiti della macchina impostiamo la massima escursione dei nostri assi, e alcune velocità non riferite alla lavorazione. Proseguendo su contatori/PCO diamo una sistemata alla grafica scegliendo l'aspetto e il tipo di visualizzazione dei contatori e se le quote sono da riferirsi all'origine del pezzo o della macchina, attenzione, selezionando mostra XYZ ad ogni passo e il Vuometro, impegnate il vostro computer ottenendo quindi una minor velocità nel flusso dati con la porta parallela, se non vi interessa vedere i numerini che cambiano disabilitateli, ne avrete un vantaggio di prestazioni massime. Nella scala di lavorazione abbiamo la possibilità di scalare il nostro disegno scegliendo se la scala vale per tutti gli assi o solo per X e Y, possiamo inoltre produrre in serie attivando la duplicazione. ISO G0 vi da la possibilità di scegliere alcuni parametri per caricare il GCode, inoltre vi può aprire la pagina per editarlo direttamente. Bene, dopo questi passaggi direi che siamo pronti e come dice un mio amico "facciamo del truciolo".
Vi consiglio di fare qualche semplice disegno e provare a far posizionare la macchina sull'origine del pezzo prima di avviare la lavorazione, iniziate con velocità molto basse, non abbandonate mai la macchina, controllate attentamente se gli assi sono liberi nel loro movimento, siate pronti a spegnere tutto.

CONCLUSIONI

Nella figura 17 vedete la fresa mentre sta lavorando durante una delle tante prove, il materiale è un tipo di espanso piuttosto compatto che si usa in edilizia come isolante termico e lo trovate per pochi euro in pannelli presso quasi tutti i centri fai-da-te, è fantastico per iniziare senza rischiare o bruciare frese, nel caso della foto la fresa ha un diametro di 3mm e sta eseguendo una passata profonda 0,2mm.
Che ve ne pare, non è un gioiellino?

Figura 17 Fase di svuotamento della parte interna passata a 0,2 mm

EMI - RFI Filters

Vitamina C:

In questa puntata verranno illustrati diversi accorgimenti utilizzabili in C per rendere l'esecuzione di un programma estremamente più veloce ed efficiente. Verrà mostrato in pratica che per ottenere delle buone prestazioni non basta utilizzare gli algoritmi giusti, ma occorre anche scrivere il codice nel modo giusto!

INTRODUZIONE

Se il nostro codice sembra troppo complesso e lento quando eseguito su un determinato processore, la soluzione non è cambiare quest'ultimo con uno più veloce, ma scrivere meglio il codice!
Questa soluzione infatti, a differenza della prima non ha praticamente nessun costo aggiuntivo, e può essere applicata sempre. Addirittura in molti casi, un codice ottimizzato può essere eseguito efficientemente anche su un processore meno potente, e quindi meno costoso. In altri casi invece una maggiore efficienza del codice può permettere di soddisfare dei vincoli temporali stringenti che non si riuscirebbero a soddisfare neanche cambiando processore. Da questo si intuisce il notevole interesse per tecniche di ottimizzazione del codice C. L'ottimizzazione del codice non è una disciplina precisa e ben codificata, è piuttosto una specie di "arte" che si impara soltanto dopo molta pratica. Per ottimizzare il codice è possibile agire a diversi livelli, e con tecniche abbastanza varie, come verrà mostrato di seguito. Quando si applicano queste tecniche però devono essere tenuti in considerazione anche gli eventuali effetti indesiderati che esse introducono. L'uso di alcune di queste tecniche spesso comporta la violazione di molte delle regole che di solito sono considerate di "buona programmazione", e questo potrebbe avere effetti negativi sia sulla leggibilità del codice, sia durante la fase di debug (è più difficile trovare e correggere gli errori). Inoltre alcune tecniche riescono ad incrementare la velocità di esecuzione a discapito della compattezza del codice. Occorre pertanto valutare bene la loro applicazione, soprattutto quando la disponibilità di memoria per memorizzare il codice non è adeguata, o comunque l'uso di memoria aggiuntiva risulta costoso.
In generale il modo in cui si ottiene un incremento della velocità di esecuzione consiste nel minimizzare il numero di istruzioni macchina da eseguire per una determinata routine, o diminuire il più possibile il numero delle operazioni "lente". Per fare questo chiaramente bisogna avere un'idea di come il compilatore traduce il nostro codice C in codice macchina, e di come si può intervenire su di esso e sul codice. Alcuni dettagli su questo aspetto sono forniti nei prossimi paragrafi.

IL COMPILATORE

La maggior parte dei compilatori offre già una serie di opzioni per incrementare le prestazioni a run-time del nostro codice. Un primo passo per ottenere codice macchina un po' più ottimizzato consiste quindi nell'abilitare queste opzioni. In questo caso non si interviene sul codice C, ma si forza il compilatore ad eseguire un lavoro più accurato e minuzioso nella fase di traduzione dal codice C in codice macchina. Il risultato di questa ottimizzazione è spesso cumulabile con quello che si può ottenere agendo manualmente sul codice. In genere è possibile selezionare un'ottimizzazione mirata alla compattezza del codice generato, oppure alla

Ottimizzazione del codice

di Antonio Di Stefano
a.distefano@farelettronica.com

velocità di esecuzione. Di solito non è possibile ottenere automaticamente entrambe, oppure è possibile solo a spese di un aumento dei tempi di compilazione. Le opzioni disponibili, ed anche la qualità dei risultati dipendono molto dal compilatore utilizzato ed anche dalla particolare architettura del processore target. In figura 1 è visibile la finestra del compilatore Dev-C++ relativa proprio alle ottimizzazioni: sono disponibili tre livelli, di complessità ed efficacia via via crescente. Le stesse opzioni si possono trovare nel noto compilatore GCC (disponibile per diversi processori e microcontrollori), utilizzando i parametri -O, -O2 e -O3.

In genere le ottimizzazioni più comuni mirano ad utilizzare il minor numero di istruzioni di salto possibile, il minor numero di accessi in memoria, o a raggruppare costanti o sottoespressioni. I primi due accorgimenti sono dovuti al fatto che le istruzioni di salto e di accesso alla memoria richiedono spesso più cicli di clock rispetto alle altre, per cui risulta vantaggioso evitarle o aggirarle, ad esempio utilizzando maggiormente i registri rispetto alla memoria per manipolare le variabili, oppure ripetere esplicitamente delle istruzioni invece di eseguire dei loop, e quindi dei salti *("loop unrolling")*. Il raggruppamento delle espressioni comuni invece permette sia di ridurre il numero di accessi in memoria, sia di eseguire una sola volta i calcoli quando si trovano espressioni identiche ripetute. Quando sono messe in pratica molte ottimizzazioni automatiche, è difficile prevedere il risultato della loro interazione, e anche se il codice prodotto in genere funzionerà in modo identico a quello non ottimizzato (a parte la differenza in velocità), può capitare che il compilatore faccia delle assunzioni sul codice sorgente, che non erano nelle intenzioni del programmatore!

Figura 1 Finestra relativa alle ottimizzazioni del compilatore Dev-C++

Un esempio classico e abbastanza illuminante a tal proposito è il seguente. Supponiamo che all'interno di un programma eseguito su un microcontrollore ad un certo punto occorra attendere la pressione di un tasto per continuare. Per fare questo si può utilizzare un *polling*, leggendo all'interno di un ciclo while la locazione di memoria che corrisponde alla porta di I/O a cui fa riferimento il pulsante. Il codice è il seguente:

```
#define KEY *((unsigned char *) 0xFF00)
…
// Attesa pressione tasto
while(!KEY) {}
…
```

La macro KEY corrisponde al contenuto della locazione di memoria 0xFF00, cioè quella in cui è mappata la porta di I/O. Supponiamo che se nessun tasto è premuto tutti i bit valgano 0. L'istruzione while utilizzata per implementare l'attesa controlla il valore di KEY, e fino a quando il valore di questa resta 0, ripete il codice tra parentesi. Quando verrà premuto un tasto la condizione non sarà più verificata, e quindi si uscirà dal loop. Nelle nostre intenzioni la locazione di memoria dovrebbe essere letta in continuazione, perché sappiamo che essa può cambiare da un momento all'altro se viene premuto un tasto. Se abilitiamo le ottimizzazioni invece il compilatore deduce dal codice che, dal momento che nessuna istruzione cambia il valore della locazione di memoria, essa rimane inalterata, e quindi inizialmente essa verrà copiata in un registro, ed i successivi confronti verranno fatti con il valore memorizza-

to nel registro! Ovviamente il valore del registro rimarrà sempre lo stesso, ed il programma non uscirà mai dal ciclo! Per evitare questa situazione occorre specificare che la locazione di memoria che stiamo leggendo può cambiare indipendentemente dal programma. Per fare questo si usa la keyword **volatile**.

```
#define KEY *((volatile unsigned char *) 0xFF00)
```

Un'altra cosa che deve essere tenuta presente è che diverse ottimizzazioni "semplici" (comunque riconducibili a quelle descritte prima) sono messe in atto normalmente dal compilatore, quindi non c'è bisogno di preoccuparsene in prima persona, anzi, si può sfruttare questo fatto per aumentare la leggibilità del codice. Ad esempio quando si usano delle espressioni aritmetiche che hanno un risultato costante, queste non vengono valutate a run-time (riducendo l'efficienza), ma il risultato viene calcolato già durante la fase di compilazione, e sostituito all'espressione. Quindi scrivere:

```
for(i=0; i<8*8*2; i++)
…
```

non risulta meno efficiente di

```
for(i=0; i<128; i++)
…
```

IL CODICE

Vediamo adesso come intervenire sul codice per aumentarne l'efficienza. Considereremo diversi accorgimenti possibili, e per ciascuno di essi verrà fornita una spiegazione.

Chiamate a funzioni

La chiamata ad una funzione nel codice C viene di solito tradotta in linguaggio macchina utilizzando delle istruzioni di salto, e delle istruzioni che accedono alla memoria per eseguire delle operazioni sullo stack per gestire gli argomenti passati o restituiti. Questi due tipi di istruzioni, come già detto, risultano di solito più lente delle altre, e quindi il loro uso dovrebbe essere limitato. È possibile fare questo in diversi modi. Una prima soluzione è quella di utilizzare delle macro al posto delle funzioni, quando possibile. Questo fa in modo che invece di richiamare una funzione, le istruzioni vengano copiare per esteso quando e dove servono. Ovviamente questo comporta un incremento della lunghezza del codice, quindi è consigliabile usare questo accorgimento per funzioni semplici, o che sono richiamate in pochi punti del codice. Ad esempio la funzione che esegue il test di un bit di una variabile:

```
int BitTest(int v, char pos)
{
  return (v>>pos)&1;
}
```

può essere vantaggiosamente definita come macro:

```
#define BitTest(v, pos)  (v>>pos)&1
```

Un altro metodo, utilizzabile con alcuni compilatori per ottenere lo stesso risultato è quello di inserire la keyword **inline** prima della funzione:

```
inline int BitTest(int v, char pos)
```

Uso dei registri

Per limitare il più possibile gli accessi in memoria, si dovrebbe fare in modo che le variabili che si utilizzano più frequentemente in una certa sezione di codice siano copiate nei registri del processore prima di eseguire le operazioni che le coinvolgono.
Per fare questo si devono innanzi tutto segnalare queste variabili al compilatore, ed in secondo luogo si deve cercare di impostare le operazioni da svolgere in una forma che utilizzi il minor numero possibile di operandi ausiliari (questo perché il numero di registri disponibili è sempre molto limitato). Per indicare che una variabile è usata di frequente (e quindi andrebbe copiata e tenuta nei registri) si aggiunge la keyword **register** alla sua dichiarazione, es.:

```
register int x;
```

Dal momento che il numero di registri è di solito limitato l'uso della keyword è considerato dal compilatore come un suggerimento più che un

comando, sarà lui a scegliere se è il caso di accettarlo o meno.
In ogni caso bisogna tenere presente che un uso eccessivo della keyword *register* può anche avere effetti negativi: se si segnalano troppe variabili come *register*, il compilatore avrà difficoltà a capire quali sono quelle per cui convenga realmente effettuare l'operazione. Oppure un numero eccessivo di variabili copiate nei registri potrebbero non lasciare posto a variabili molto più critiche, ma meno "visibili".

Uso delle variabili globali

Il passaggio dei parametri alle funzioni è di solito realizzato tramite lo stack, cioè le variabili sono memorizzate nello stack prima di chiamare la funzione. L'uso dello stack comporta un aumento degli accessi in memoria, e quindi può rallentare le operazioni. Una soluzione semplice a questo inconveniente è quello di usare delle variabili globali, anziché passarle come argomenti. Ricordiamo che le variabili globali sono delle variabili definite fuori da qualsiasi funzione (compreso il main), e per questo sono visibili da qualsiasi punto del programma. Può essere vantaggioso quindi creare un set di variabili globali quando queste siano utilizzate come parametro da molte funzioni, o si riferiscano a dei dati effettivamente condivisi dalle funzioni. Comunque l'uso delle variabili globali di solito è sconsigliato, perché può creare dei problemi di coerenza e può diminuire la leggibilità del codice. Inoltre anche il debug del programma risulta più difficoltoso: se si verifica un errore sui dati più essere più difficile capire chi o cosa lo ha originato (è anche vero comunque che proprio la loro visibilità le rende più osservabili).

Uso del goto

Un'altra pratica che di solito è sconsigliata è quella di utilizzare l'istruzione **goto**. Questa istruzione può creare gli stessi problemi visti prima a proposito delle variabili globali, ma risulta molto utile per velocizzare e semplificare il codice in molti casi in cui sono coinvolti complicati controlli o sequenze di codici simili ripetute. Anche se come già detto i salti in sé sono delle istruzioni piuttosto lente, le semplificazioni introdotte dal goto possono dare un vantaggio tale da rendere conveniente il suo utilizzo. Ad esempio consideriamo il seguente codice:

```
k=0;
for(i=0; i<100; i++) {
  for(j=24; j<200; j++) {
    if (w[j]==0) goto esci;
    k=k+v[i]/w[j];
  }
}
esci:
```

Sono stati utilizzati due cicli for annidati per eseguire un calcolo iterativo che prevede una divisione. Se il denominatore viene trovato uguale a 0 occorre interrompere il calcolo. Per uscire direttamente dai due cicli for è stato utilizzato un goto. Se non si fosse fatto così sarebbe stato necessario eseguire due controlli, uno in ciascuno dei due loop, che sarebbero stati eseguiti ad ogni iterazione. Si possono presentare dei casi molto più intricati di questo, in cui si possono risparmiare anche molti più controlli. L'uso del goto in qualche modo permette di utilizzare uno stile di programmazione simile a quello che si utilizza con l'assembly.

Dati, tipi e strutture

Per rendere il più possibile veloce l'esecuzione di operazioni aritmetiche, logiche o di movimento è consigliabile utilizzare il più possibile dei tipi di dati della lunghezza ottimale per la macchina su cui verrà eseguito il codice. Ad esempio molti microcontrollori ad 8 bit possono operare direttamente soltanto su dati ad 8 bit (ad esempio alcuni PIC), altri possono operare su dati sia da 8 che da 16 (es. gli 8051/2), mentre un microprocessore a 32 bit può quasi sempre gestire direttamente dati da 8, 16, e 32 bit, anche se magari esegue le moltiplicazioni in un ciclo di clock solo su 8 bit (es. gli ARM7).
Utilizzare dati di larghezza maggiore di quella richiesta può comportare un certo rallentamento, ed inoltre può non permette l'uso dei registri per memorizzare le variabili. In ogni caso, indipendentemente dalla larghezza dei dati, è comunque importante cercare di utilizzare i tipi nativi offerti dall'ANSI C, in caso contrario si è costretti ad eseguire del codice aggiuntivo per ogni operazione da svolta sui dati, aumentando

molto il tempo di esecuzione dell'operazione, ed il numero di accessi in memoria. Una delle ragioni dell'inefficienza che si può originare quando si gestiscono dei tipi non standard in C, è dovuto al fatto che risulta complicato reperire delle informazioni che invece in assembly sarebbero a disposizione automaticamente. Ad esempio, per implementare un'addizione a 64 bit, occorrerebbe conoscere il valore del flag di riporto del processore, che non è accessibile direttamente dal C. Occorre quindi calcolarlo esplicitamente, aumentano (inutilmente) il numero di istruzioni da svolgere.
Un discorso a parte meritano i tipi strutturati. Essi sono implementati in maniera differente da ogni compilatore, anche in relazione all'architettura hardware sottostante. Questo implica che non sempre la selezione di un campo si traduce in una sola lettura dalla memoria, ma può comportare anche operazioni aggiuntive come shifting multipli e mascheratura. Questo si verifica in particolare quando la dimensione dei campi non coincide con la granulosità d'indirizzamento del microprocessore utilizzato. Consideriamo ad esempio l'utilizzo di campi di bit:

```
typedef struct {
  char ID : 4;
  char Tipo : 5;
  short Lunghezza : 12;
} header;
```

Per ciascuno dei campi della struttura è stata definita la lunghezza in bit (questa è una possibilità poco conosciuta dell'ANSI C). Se il compilatore memorizza i campi in sequenza, per leggere campo ID sarà necessaria una lettura ed una mascheratura (un AND con 0x0F), per leggere il campo Tipo invece, sarà necessaria la lettura di ben due byte, lo shifting di 4 posizioni a destra, e la mascheratura. Alcuni compilatori in realtà non memorizzano sequenzialmente i bit, ma pongono i vari campi in locazioni indirizzabili singolarmente, lasciando quindi un po' di spazio "vuoto" tra loro. In questo caso non sarà necessario lo shifting, ma soltanto la mascheratura. Una situazione analoga si può verificare anche utilizzando i normali tipi, quindi è sempre il caso di valutare anche questo aspetto prima di utilizzare un certo tipo di dati.

Operazioni aritmetiche

Normalmente le operazioni aritmetiche non richiedono tutte lo stesso tempo di esecuzione, in genere l'addizione e la sottrazione possono essere eseguite in un solo ciclo di clock, la moltiplicazione può richiedere più di un ciclo, mentre la divisione (o anche il "modulo" %), anche quando supportata dall'hardware, ne richiede sempre un numero abbastanza grande (decine). Da questo si intuisce che quando possibile la divisione deve essere evitata, e l'uso della moltiplicazione deve essere ridotto.
In molti casi questo non è difficile, e si può ottenere utilizzando qualche piccola accortezza, come mostrato nel seguente esempio, dove si deve scrivere un valore in una matrice bidimensionale di dimensioni 30x20, mappata in un'area di memoria lineare (ad esempio uno schermo a caratteri). In questo caso la difficoltà nasce dal fatto che le prime 10 colonne devono essere lasciate inalterate, quindi non è possibile soltanto incrementare il puntatore, ma bisogna utilizzare un "indirizzamento" riga-colonna:

```
for(j=0; j<20; j++) {
  for(i=10; i<30; i++) {
    a[i+30*j]=79;
  }
}
```

Per ogni iterazione viene utilizzata una somma ed una moltiplicazione, per un totale di 20*30 operazioni solo per calcolare il valore dell'indice! Riorganizzando il ciclo si può evitare l'uso della moltiplicazione:

```
k=0;
for(j=0; j<20; j++) {
  for(i=10; i<30; i++) {
    a[i+k]=79;
  }
  k+=30;
}
```

Un altro modo per evitare le moltiplicazioni o le divisioni, applicabile quando si ha a che fare con potenze di due, è quello di utilizzare le operazioni di bit shifting (<< e >>).
Uno scorrimento a destra equivale ad una divisione per 2, a sinistra ad una moltiplicazione per

2. Scorrendo più bit si possono ottenere le altre potenze di 2:

```
a = b*32;
```

equivale a:

```
a = b<<5;
```

Gli scorrimenti di una posizione sono eseguiti normalmente in un solo ciclo di clock, quindi più velocemente delle moltiplicazioni. Quelli di più posizioni possono richiedere tanti cicli quanti sono i posti da scorrere (a meno che il processore non disponga di un *barrel shifter*, in questo caso è necessario un solo ciclo di clock). Se il processore non è dotato di un moltiplicatore hardware allora risulta sempre conveniente usare gli scorrimenti, se ne è dotato allora occorre valutare bene quale dei due metodi risulti più vantaggioso.
Per quanto riguarda la divisione o altre operazioni intrinsecamente lente (operazioni trigonometriche, numeri casuali, funzioni complesse…) è possibile spesso utilizzare delle *look-up tables*, cioè delle tabelle che contengono i risultati precalcolati. In questo modo sarà possibile ottenere un risultato semplicemente leggendolo dalla tabella.
Ovviamente questo metodo incrementa la dimensione del codice oggetto, ma può accelerare notevolmente l'esecuzione del programma, ed è particolarmente adatto in quei casi in cui il numero di elementi della tabella si conosce con precisione a priori (es. calcolo della FFT). Inoltre, come già detto, per ottenere delle prestazioni accettabili nella maggior parte dei casi è necessario rinunciare all'uso dell'aritmetica floating point, e sostituirla con quella fixed point (si vedano le sue puntate precedenti per maggiori dettagli sull'argomento).

Librerie standard

Al contrario di quello che si potrebbe pensare, non sempre le librerie standard, messe a disposizione dai compilatori, implementano nel modo più efficiente le funzioni a cui sono preposte. Questo è dovuto, oltre che alle differenze d'implementazione da un compilatore all'altro, anche al fatto che le funzioni messe a disposizione cercano di gestire i casi più generali.
Questo comporta non solo una potenziale inefficienza (o meglio, non ottimizzazione), ma anche un incremento di dimensioni del nostro codice oggetto, spesso non necessario. In molti casi quindi è consigliabile riscrivere alcune funzioni in maniera molto più sintetica ed ottimizzata per la particolare applicazione.
Consideriamo ad esempio la funzione *memcpy* (libreria strings.h), che serve per copiare un blocco di dati da un'area di memoria ad un'altra. Essa in genere legge e scrive la memoria a byte.Se il processore è dotato di un indirizzamento a 32 bit, è possibile copiare la stessa quantità di memoria 4 volte più velocemente! (Copiando word da 32 bit invece che singoli byte). Oppure se nel sistema è presente un DMA, è molto più conveniente scrivere una versione della memcpy che lo utilizza per effettuare le operazioni di copia, magari utilizzando l'inline assembler.

Uso dell'inline assembler

Come si può facilmente immaginare il modo migliore per ottenere un programma il più efficiente e compatto possibile, consiste nello scriverlo manualmente in assembly. Di solito questo procedimento porta ad utilizzare il minor numero possibile di istruzioni, ed anche il minor numero di risorse. Nessun compilatore (attualmente) è capace di raggiungere un simile livello di ottimizzazione.
Anche se si è scelto di utilizzare il linguaggio C per scrivere il programma, è comunque possibile utilizzare l'assembly e le sue potenzialità per scrivere soltanto le parti più lente o computazionalmente più pesanti del programma.
Queste parti possono essere inglobate nel codice C in modo molto semplice: è sufficiente infatti utilizzare la keyword **asm**, seguita dal blocco di codice assembler, come mostrato nell'esempio seguente:

```
void SetVideoMode(void)
{
   asm
   {
      mov ax,0x13
```

```
        int 0x10
    }
}
```

Questa piccola funzione ha lo scopo di richiamare la modalità video 0x13 (grafica) su un PC [attenzione: il sistema operativo potrebbe non accettare di buon grado una simile funzione!]. Il codice assembly è costituito soltanto da un paio di istruzioni per il processore x86, ed il tutto è stato racchiuso in una funzione.
Quest'ultimo particolare non è strettamente necessario, anche se in alcuni casi può facilitare ad esempio il passaggio di parametri. Quando si usano dei blocchi di codice assembly bisogna tenere presente che occorrerebbe prima salvare il valore dei registri su cui si andrà ad operare, e ripristinarlo all'uscita del blocco (il compilatore non lo fa automaticamente). Inoltre bisogna ricordare che l'uso del codice assembly, essendo strettamente specifico per una certa macchina, rende l'intero codice difficilmente portabile.
Tuttavia proprio la specificità dell'hardware può essere uno delle ragioni principali per utilizzare l'inline assembler. Infatti alcune caratteristiche peculiari di alcune architetture (ad esempio la presenza di alcuni tipi di co-processori) non sono normalmente sfruttate dai compilatori, ed è quindi necessario scrivere manualmente delle funzioni che le utilizzino per accelerare l'esecuzione, e questo può essere fatto spesso solo in assembly.
In questi casi è consigliabile porre queste funzioni in un modulo a parte, in modo da poterlo facilmente sostituire o modificare se si cambia hardware.

Costrutti switch

Il costrutto switch è utilizzato di solito quando in un programma è necessario scegliere quale sezione di codice eseguire in base al valore di un parametro. Molti compilatori traducono questo costrutto con delle istruzioni di confronto e salto condizionato, poste sequenzialmente prima di ciascuno dei casi elencati. Questo implica che prima di raggiungere l'ultimo caso devono essere eseguite tutte le istruzioni di confronto e salto relative agli altri casi, con un note-

vole incremento del tempo di esecuzione.
Per limitare questo effetto è possibile procedere in diversi modi. Un primo accorgimento è quello di sistemare prima i casi che si verificheranno più frequentemente (con più probabilità), e di seguito gli altri. In questo modo i casi più frequenti richiederanno l'esecuzione di un numero minore di controlli, e verranno eseguiti in minor tempo.
Altri accorgimenti consistono nell'uso ripetuto dell'istruzione goto, o nella creazione di una tabella dei salti (anche se quest'ultimo metodo risulta decisamente più complesso da mettere in pratica in generale).

Ulteriori consigli

Un'altra regola che bisognerebbe tenere sempre presente quando si vuole scrivere un codice molto efficiente è evitare assolutamente l'uso di tecniche di ricorsione. Questa infatti comporta un uso massiccio dello stack e quindi incrementa notevolmente il numero di accessi in memoria. Inoltre proprio per il fatto che lo stack può crescere molto, si potrebbero verificare casi di stack overflow, cioè esaurimento dello spazio ad esso riservato.
Questo è ancora più probabile quando assieme a funzioni ricorsive si utilizzano le interruzioni. Tra l'altro anche le stesse interruzioni (di cui si parlerà in un prossimo articolo), al contrario di quello che si potrebbe credere, possono appesantire l'esecuzione del codice.
Di solito infatti le routine di servizio delle interruzioni come prima operazione salvano il contenuto dei registri prima di iniziare le elaborazioni. Questo "context switch", se ripetuto frequentemente può costituire un notevole overhead. Nei casi in cui ci siano delle interruzioni molto frequenti, ma limitate nel tempo, può essere conveniente gestirle attraverso un polling, magari periodico anziché continuo.
Un'altra osservazione interessante riguarda i loop realizzati con i cicli for. Normalmente un ciclo del tipo:

```
for(i=0; i<10; i++) {…}
```

viene tradotto in codice macchina come:

```
- esegui istruzioni del ciclo
- incrementa i
- confronta i e 10
- se diverso ripeti loop
```

Per controllare la fine del ciclo viene eseguita un'istruzione di *compare*, e poi un salto condizionato.
Se invece si scrivere il ciclo utilizzando un conteggio decrescente, cioè:

```
for(i=10; i!=0; i--) {…}
```

si ottiene un codice macchina simile al seguente:

```
- esegui istruzioni del ciclo
- decrementa i
- se i diverso da zero ripeti il loop
```

In questo caso non è stato necessario utilizzare l'istruzione di compare, dal momento che la maggior parte dei processori dispongono di un'istruzione di salto condizionata dal un flag di zero o non zero.
Anche se si tratta di una sola istruzione in meno, questa è eseguita ad ogni iterazione, quindi il tempo "sprecato" è dato dal tempo richiesto per eseguire l'istruzione di *compare*, per il numero di iterazioni (solo 10 in questo caso).
Quando possibile quindi è conveniente invertire la direzione del conteggio dei loop.

CONCLUSIONE

Per scrivere del codice molto ottimizzato è necessaria una certa esperienza, ed anche una buona conoscenza del processore che si sta utilizzando, nonché del compilatore. Se ben applicati comunque, i suggerimenti visti in questa puntata possono rendere l'esecuzione del codice molto più veloce (anche oltre 10 volte!).
Però come già spiegato, questo incremento di prestazioni si paga in termini di aumento di dimensioni del codice oggetto, scarsa leggibilità e portabilità, e maggiore difficoltà durante le fasi di debug.
Sta all'esperienza del programmatore decidere se ed in che misura utilizzare queste tecniche.
La prossima puntata tratterà un argomento che è stato accennato in questa: le tecniche di debug del codice.

Rispondi correttamente al quiz e potrai vincere un abbonamento omaggio a

Fare elettronica

Partecipare è semplicissimo:

rispondi al quesito seguendo il regolamento e, se la risposta si rivelerà esatta, potrai vincere un abbonamento omaggio (o il rinnovo qualora fossi già abbonato) a Fare Elettronica. Ogni mese sulle pagine della rivista troverai la soluzione del quesito del mese precedente e il nome del vincitore di uno dei quesiti pubblicati. Per tutti i partecipanti che risponderanno utilizzando l'apposito modulo su www.farelettronica.com/lesotutte, è previsto comunque un coupon del 10% di sconto utilizzabile per un acquisto sullo shop di www.farelettronica.com.

QUESITO codice LST24307

Si dispone di un 74SN00 e di un 74LS00 ovvero due circuiti integrati contenenti porte NAND TTL: il primo nella versione standard, il secondo nella versione LS (basso consumo e alta velocità). I parametri tipici delle due famiglie di porte sono riportati nella tabella ed il loro significato è il seguente: V_{OLmax} è la massima tensione di uscita sul livello basso, V_{OHmin} è la minima tensione di uscita sul livello alto, V_{IHmin} è la minima tensione di ingresso riconosciuta come livello alto, V_{ILmax} è la massima tensione di ingresso riconosciuta come livello basso, I_{IHmax} è la massima corrente di ingresso sul livello alto, I_{ILmax} è la massima corrente di ingresso sul livello basso, I_{OHmax} è la massima corrente di uscita sul livello alto e I_{OLmax} è la massima corrente di uscita sul livello basso.

	V_{OLmax}	V_{OHmin}	V_{IHmin}	V_{ILmax}	I_{IHmax}	I_{ILmax}	I_{OHmax}	I_{OLmax}
Standard	0.5V	2.7V	2V	0.8V	20mA	400mA	400mA	8mA
LS	0.4V	2.4V	2V	0.8V	40mA	1.6mA	0.4mA	16mA

Usando una porta standard per pilotare più porte LS, quale è il numero massimo N di porte LS pilotabili correttamente?

Scadenza: il termine ultimo per rispondere è il 30 Settembre 2005

Regolamento

1) Il quiz è aperto a tutti i lettori.
2) Saranno considerate esclusivamente le risposte pervenute entro la scadenza indicata nel quesito.
3) Inviare la risposta compilando il modulo su www.farelettronica.com/lesotutte oppure via **fax** allo 02/66508225 o tramite **lettera** a Inware Edizioni, via Cadorna 27- 20032 Cormano (MI), citando il codice del quesito.
4) Indicare nella risposta il tuo nome, cognome, indirizzo, telefono ed eventuale indirizzo email. Non ci sarà possibile considerare le risposte con **dati mancanti o insufficienti.**

Sì approfitto della fantastica occasione e mi abbono a 11 numeri di Fare Elettronica e pagherò solo € 45,00

DATI PERSONALI

Nome . . . .

Cognome . . . .

Via . . . . n° . . . .

Cap . . . . Città . . . . Prov . . . .

Tel . . . . Fax . . . .

Email . . . .

Ragione Sociale . . . .

P.Iva . . . . ❍ Ricevuta

Privacy. Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni Via Cadorna 27 – 20032 Cormano o trmite email a info@inwaredizioni.it

ABBONATI SUBITO

Compila, ritaglia e spedisci via fax questo coupon allo **02-66508225**

Spedisci questa pagina in busta chiusa a:
INWARE Edizioni
Via Cadorna, 27/31 - 20032 Cormano (MI)

Chiamaci allo **02-66504794**

Abbonati on-line sul sito
www.farelettronica.com/abbonamento

MODALITÀ DI PAGAMENTO

❍ **CARTA DI CREDITO**

❑ American Express ❑ Visa ❑ Mastercard

Titolare . . . .

n° . . . . scad . . . .

❍ **VERSAMENTO SUL CCP 22790232**

Allegare la ricevuta (o copia) del versamento intestato a Inware Srl, indicando nella causale: "Abbonamento Fare Elettronica"

❍ **BONIFICO BANCARIO**

Appoggiarlo su: **Poste Italiane - CIN: Z - ABI: 07601 CAB: 01600 - C/C: 000022790232** intestato ad **Inware srl**

❍ **ALLEGO UN ASSEGNO**

intestato a Inware Srl

Firma . . . .

Abbonati subito!

- ✔ **Risparmierai** ben 15,50 euro sul prezzo di copertina
- ✔ Avrai la garanzia del **prezzo bloccato** per un anno
- ✔ La rivista ti sarà **recapitata comodamente** a casa
- ✔ Compreso con l'abbonamento (o il rinnovo) riceverai un **buono sconto del 20%** per un tuo prossimo acquisto sul sito www.farelettronica.com e, insieme alla merce ordinata, ti sarà recapitato l'esclusivo portapenne di Fare Elettronica

INWARE EDIZIONI